



UNIVERSIDAD  
DE LA REPÚBLICA  
URUGUAY



FACULTAD DE  
INGENIERÍA

# Exploración evolutiva del espacio latente de redes neuronales antagónicas para la generación de rostros humanos

Informe de Proyecto de Grado presentado por

Juliana Caballero, Julieta Caffera y Gonzalo Piria

en cumplimiento parcial de los requerimientos para la graduación de la carrera  
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de  
la República

Supervisor

Sergio Nesmachnow

Montevideo, 8 de octubre de 2024



Exploración evolutiva del espacio latente de redes neuronales antagonicas para la generaci3n de rostros humanos por Juliana Caballero, Julieta Caffera y Gonzalo Piria tiene licencia [CC Atribuci3n 4.0](https://creativecommons.org/licenses/by/4.0/).

# Agradecimientos

En esta fase final de nuestras carreras, nos gustaría expresar nuestro profundo agradecimiento a todos aquellos que hicieron posible este proyecto.

En primer lugar les agradecemos a nuestras familias que siempre nos han brindado su apoyo incondicional para poder cumplir todos nuestros objetivos personales y académicos.

Queremos agradecer también a los muchos docentes que han sido parte de nuestro camino universitario, por habernos transmitido los conocimientos necesarios para llegar hasta aquí. Especialmente agradecemos a Sergio Nesmachnow, quien nos brindó orientación constante y acompañamiento a lo largo de todo el proyecto.

Asimismo, extendemos nuestro agradecimiento al equipo de Cluster UY. La realización de todos los experimentos presentados en este estudio fue posible gracias a su infraestructura y al continuo soporte técnico proporcionado por su equipo.

¡Gracias a todos por su colaboración y apoyo!



# Resumen

El proyecto se centra en la generación de rostros humanos sintéticos utilizando un modelo de Red Generativa Antagónica, abordando la necesidad de crear imágenes realistas y libres de sesgos. El objetivo principal es desarrollar estrategias evolutivas para explorar el espacio latente de las redes generativas antagónicas y generar rostros con atributos específicos de raza y género. Además, se busca garantizar que estos rostros sean reconocidos por el analizador DeepFace como la misma persona, a pesar de las variaciones en raza y género. Se implementa un algoritmo evolutivo multi-objetivo que guía la generación de rostros humanos sintéticos evaluando la precisión de los resultados. El algoritmo utiliza en su función de evaluación un generador de rostros, StyleGAN3, y un analizador, DeepFace. El analizador mide la similitud entre los rostros generados y una imagen objetivo, así como la correspondencia con los atributos de raza y género deseados. Los resultados indican que StyleGAN3 y DeepFace presentan un sesgo hacia los rostros de individuos blancos, adultos y masculinos, lo que se refleja en la predominancia de estos perfiles y una mayor precisión en su evaluación. Sin embargo, el algoritmo evolutivo demostró su capacidad para generar rostros con atributos específicos de raza, género y capaces de engañar a DeepFace, produciendo resultados efectivos sin importar el objetivo.

**Palabras clave:** Reconocimiento facial, Generación de rostros sintéticos, StyleGAN3, Algoritmos evolutivos, DEAP, DeepFace, Sesgo



# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco teórico</b>	<b>3</b>
2.1. Algoritmos evolutivos . . . . .	3
2.1.1. Funcionamiento general de los algoritmos evolutivos . . . . .	3
2.1.2. Algoritmos genéticos . . . . .	4
2.1.3. Principales componentes de un algoritmo evolutivo . . . . .	5
2.1.4. Algoritmos evolutivos multi-objetivo . . . . .	8
2.2. Redes generativas antagónicas . . . . .	10
<b>3. Revisión de antecedentes</b>	<b>15</b>
3.1. Motivación y metodología . . . . .	15
3.2. Uso de GANs y algoritmos evolutivos para explorar el espacio latente de GANs . . . . .	15
3.3. Generación de rostros a través del uso de algoritmos evolutivos para la exploración del espacio latente de GANs . . . . .	19
3.4. Utilidad de los artículos relevados . . . . .	23
<b>4. Descripción del problema e implementación de la solución</b>	<b>25</b>
4.1. Descripción del problema . . . . .	25
4.2. Modelado de la solución . . . . .	26
4.2.1. Representación de los individuos . . . . .	27
4.2.2. Función de evaluación o fitness . . . . .	27
4.2.3. Selección . . . . .	28
4.2.4. Operadores de variación . . . . .	29
4.2.5. Población . . . . .	29
4.3. Arquitectura del pipeline . . . . .	30
4.3.1. Estructura y funcionamiento general del pipeline . . . . .	30
4.3.2. Funcionamiento de los principales componentes del pipeline . . . . .	31
4.3.3. Eliminación de la capa softmax de DeepFace para el género . . . . .	36
<b>5. Experimentación</b>	<b>47</b>
5.1. Metodología para el análisis experimental . . . . .	47

5.1.1.	Metodología para la selección del modelo de detección y el modelo de análisis . . . . .	47
5.1.2.	Metodología para el análisis de la exploración del espacio de vectores de entrada de StyleGAN3 . . . . .	48
5.1.3.	Metodología para el análisis de sesgo en la generación y evaluación de rostros . . . . .	48
5.1.4.	Elección de rostros para la experimentación . . . . .	50
5.1.5.	Metodología para la elección del porcentaje de expansión de imágenes . . . . .	52
5.1.6.	Metodología para la elección de la configuración paramétrica . . . . .	52
5.1.7.	Metodología para el cálculo experimental del umbral . . . . .	55
5.1.8.	Metodología para la validación . . . . .	56
5.1.9.	Ambiente de desarrollo y ejecución . . . . .	57
5.2.	Resultados del análisis experimental . . . . .	58
5.2.1.	Resultados del análisis de desempeño de los modelos de detección y análisis . . . . .	58
5.2.2.	Resultados del análisis de la exploración del espacio de vectores de entrada de StyleGAN3 . . . . .	60
5.2.3.	Resultados del análisis de sesgo en la generación y evaluación de rostros . . . . .	62
5.2.4.	Resultados de la elección del porcentaje de expansión de imágenes . . . . .	70
5.2.5.	Resultados de la configuración paramétrica . . . . .	73
5.2.6.	Resultados del cálculo experimental del umbral . . . . .	75
5.2.7.	Resultados de la validación . . . . .	77
<b>6.</b>	<b>Conclusiones y trabajo futuro</b>	<b>87</b>
6.1.	Conclusiones . . . . .	87
6.2.	Trabajo futuro . . . . .	89
<b>A.</b>	<b>Tablas de configuración paramétrica</b>	<b>95</b>
A.1.	Configuración paramétrica del algoritmo . . . . .	95
A.1.1.	Test de rangos Friedman . . . . .	95
A.1.2.	Resultados de configuración paramétrica . . . . .	97
<b>B.</b>	<b>Recorte de imágenes</b>	<b>103</b>
B.1.	Análisis de expansión de imágenes . . . . .	103
<b>C.</b>	<b>Validación</b>	<b>107</b>
C.1.	Evolución del hipervolumen . . . . .	107
C.2.	Evolución del frente de Pareto . . . . .	110
C.2.1.	Instancia I1 . . . . .	110
C.2.2.	Instancia I4 . . . . .	112
C.2.3.	Instancia I6 . . . . .	114
C.2.4.	Instancia I7 . . . . .	116
C.2.5.	Instancia I8 . . . . .	118

# Capítulo 1

## Introducción

El reconocimiento facial ha emergido como una tecnología fundamental en diversas áreas, abarcando desde la seguridad y la vigilancia hasta la autenticación biométrica y la personalización de servicios. Su versatilidad y utilidad han sido evidentes en aplicaciones cotidianas, como el desbloqueo de dispositivos móviles o el acceso a instalaciones protegidas (Mohsienuddin y Sabri, 2020).

Ha surgido a su vez un interés creciente en la investigación sobre la generación de rostros humanos sintéticos en el campo del reconocimiento facial y la inteligencia artificial. A medida que las tecnologías de reconocimiento facial avanzan, surge la necesidad de desarrollar métodos más eficientes y precisos para generar imágenes sintéticas que sean realistas y libres de sesgos. La capacidad de generar imágenes sintéticas realistas es esencial para asegurar la precisión y equidad en el desarrollo y la evaluación de sistemas automáticos de reconocimiento facial, así como para abordar inquietudes éticas relacionadas con el sesgo y la privacidad de los datos. Por consiguiente, la exploración de nuevas estrategias para la generación controlada de rostros sintéticos se posiciona como una prioridad en la evolución de la inteligencia artificial y el reconocimiento facial.

El proyecto busca generar rostros humanos sintéticos con atributos específicos de manera eficiente y contribuir al avance de la tecnología de reconocimiento facial, promoviendo la investigación en áreas clave como la computación evolutiva y las redes neuronales generativas (GANs). El proyecto se centra en el estudio, diseño, implementación y validación de estrategias evolutivas para la exploración del espacio latente de GANs con el objetivo de generar muestras sintéticas que cuenten con un conjunto de características o atributos deseados y sean a la vez similares a una imagen objetivo. Se aborda el problema de generación y análisis de rostros humanos, trabajando con algoritmos y repositorios de datos especializados en el tema (Serengil y Ozpinar, 2020).

El proyecto plantea la implementación de un algoritmo evolutivo como solución al problema de la exploración del espacio latente de las GANs. El objetivo principal es desarrollar estrategias para guiar la generación de rostros humanos mediante la exploración del espacio de codificaciones de atributos, lo que permitirá la creación de conjuntos de datos con características deseadas como

raza y género, que a su vez puedan engañar a DeepFace. Siguiendo la línea de investigación de [Machín y cols. \(2021\)](#), quienes propusieron explorar el espacio latente a través de algoritmos evolutivos para realizar la búsqueda de manera más eficiente y con menor demanda de capacidad de cómputo, se desarrolla y valida un algoritmo evolutivo multi-objetivo para el problema de la generación sintética de rostros humanos con determinados atributos, y se explora el uso de técnicas de computación de alto desempeño para acelerar estos algoritmos.

Se identifica que ciertos modelos, como StyleGAN3 y DeepFace, presentan sesgos en los datos utilizados para su entrenamiento, lo que impacta la calidad y la equidad de las imágenes generadas. El sesgo de StyleGAN3 y DeepFace puede manifestarse en una representación desproporcionada de características como género, raza y edad, afectando la diversidad y la precisión de los resultados obtenidos.

Las contribuciones del proyecto incluyen un análisis del problema de exploración del espacio latente de las GANs y su aplicación en la generación de rostros humanos sintéticos, a través de un relevamiento y análisis de trabajos relacionados en el campo. Además, se diseña e implementa un algoritmo evolutivo multi-objetivo con el fin de generar, con la utilización de una GAN, rostros sintéticos de alta calidad. Todos los algoritmos y experimentos descritos en el proyecto fueron implementados y validados en la plataforma del Centro Nacional de Supercomputación (Cluster-UY), utilizando casos de estudio creados a partir de datasets públicos existentes como UTKFace ([Zhang y cols., 2017](#)). Finalmente, se realizó un análisis exhaustivo de la calidad de los resultados y el rendimiento computacional de las implementaciones desarrolladas.

En síntesis, se brinda la implementación de un algoritmo evolutivo multi-objetivo diseñado para generar rostros realistas con determinados atributos de raza y género, que puedan engañar a DeepFace. El algoritmo desarrollado logró abordar los tres objetivos distintos simultáneamente. Se generaron imágenes que no solo se asemejaban visualmente a la imagen objetivo, sino que también cumplen con los atributos específicos de raza y género. Además, estas imágenes fueron reconocidas como la misma persona por el sistema DeepFace, demostrando la efectividad del enfoque en lograr engañar al reconocedor.

El presente documento se encuentra estructurado como se explica a continuación. El [Capítulo 2](#) presenta el marco teórico, abordando conceptos fundamentales como los algoritmos evolutivos, las redes neuronales generativas antagónicas y las distintas herramientas y bibliotecas utilizadas en el proyecto. El [Capítulo 3](#) plantea el estudio de antecedentes respecto al problema de la exploración del espacio latente de las GANs y el uso de algoritmos evolutivos como solución. El [Capítulo 4](#) define el proceso de implementación de la solución presentada en el proyecto, junto con la experimentación inicial de sesgo. En el [Capítulo 5](#) se presenta la experimentación sobre la solución planteada, detallando la elección de rostros a utilizar, las configuraciones elegidas y los resultados obtenidos. Finalmente se plantean en el [Capítulo 6](#) las conclusiones del proyecto y trabajo futuro.

# Capítulo 2

## Marco teórico

En este capítulo se presenta el marco teórico que proporciona conceptos e ideas claves respecto a algoritmos evolutivos genéticos y redes neuronales generativas antagonicas. El marco teórico permitirá establecer una base sólida para la formulación de las hipótesis y el diseño metodológico del proyecto.

### 2.1. Algoritmos evolutivos

En esta sección se detalla el funcionamiento de los algoritmos evolutivos, así como el de una de sus variantes, los algoritmos genéticos.

#### 2.1.1. Funcionamiento general de los algoritmos evolutivos

Los algoritmos evolutivos son un tipo de algoritmo de optimización y búsqueda que se inspira en los principios de la evolución natural para resolver problemas complejos. Se basan en la idea de que, mediante la aplicación de operadores evolutivos como la selección, el cruce y la mutación, se puede encontrar una solución óptima o aproximada a un problema dado (Holland, 1975; Whitley, 2001).

Un algoritmo evolutivo puede tener muchas variantes, pero todos siguen la misma idea: Dada una población inicial de representaciones candidatas (individuos) que se genera aleatoriamente o a través de un proceso de inicialización, se establece una competencia de selección natural a través de varias generaciones basada en la supervivencia del más apto (Joglekar y Tungare, 2001). Cada individuo representa una solución potencial al problema en cuestión (Koza y Poli, 1997). A lo largo de múltiples generaciones, los individuos con características favorables tienden a ser seleccionados y reproducidos. Por otro lado los menos aptos tienen menos probabilidades de contribuir a la siguiente generación. El proceso de evolución de individuos se repite hasta que se alcanza una condición de terminación predefinida, como un número máximo de generaciones o la consecución de una solución satisfactoria (Forrest, 1996).

La evaluación de calidad de los individuos se realiza mediante una función de fitness. La función de fitness asigna un valor de aptitud a cada individuo. Dependiendo del objetivo del algoritmo, ya sea de minimización o de maximización, un menor o mayor valor retornado por la función de fitness indica una mejor solución. Sobre la base de los valores de aptitud, se eligen en base a un operador de selección algunos de los mejores individuos para generar la siguiente generación. El proceso de evolución busca causar un aumento en la aptitud general de la población, con el objetivo de acercarse cada vez más a una solución óptima (Deb, 2004).

Dentro del proceso de evolución los individuos son sometidos a operadores evolutivos como el cruce y la mutación (Deb, 2004). El cruce es un operador que se aplica a dos o más individuos seleccionados, a los que se les denomina padres, produciendo uno o más nuevos candidatos, denominados hijos. El proceso de cruce busca combinar las características fuertes de los padres para formar individuos con un mejor fitness. La mutación se aplica a un individuo y resulta en un nuevo candidato, introduciendo variabilidad. La mutación es crucial para aumentar la diversidad de la población y evitar caer en mínimos locales, asegurando una exploración más completa del espacio de soluciones.

Los algoritmos evolutivos son una alternativa a métodos como el de descenso por gradiente, ya que son capaces de no quedar atrapados en mínimos locales.

En el [Algoritmo 1](#) se muestra el código genérico para un algoritmo evolutivo.

---

**Algoritmo 1** Pseudocódigo del funcionamiento de un algoritmo evolutivo

---

```
inicio  
inicializar la población con soluciones candidatas aleatorias;  
evaluar cada candidato;  
repetir  
    seleccionar a los padres;  
    recombinar parejas de padres;  
    mutar la descendencia resultante;  
    evaluar a los nuevos candidatos;  
    seleccionar a los individuos para la siguiente generación;  
hasta que se satisfaga la condición de terminación  
fin
```

---

### 2.1.2. Algoritmos genéticos

Dentro de la amplia variedad de algoritmos evolutivos, se encuentran los algoritmos genéticos. Los algoritmos genéticos representan a los individuos mediante estructuras simples que se asemejan a cromosomas. Son utilizados fundamentalmente en la resolución de problemas de búsqueda y optimización (Ilachinsky, 1997).

Los algoritmos genéticos operan mediante la manipulación de cromosomas. La forma en que se codifican los cromosomas suele variar según el problema que se busca resolver. Un cromosoma está compuesto por un conjunto de genes, donde cada gen representa una característica específica del individuo y está ubicado en una posición particular del cromosoma, conocida como locus. Los distintos valores que puede tomar un gen se denominan alelos.

### 2.1.3. Principales componentes de un algoritmo evolutivo

En esta sección se desarrollan los principales componentes de un algoritmo evolutivo.

#### Representacion de los individuos

En el contexto de los algoritmos evolutivos, se busca generar un mapeo entre el conjunto de posibles soluciones del problema original, conocidas como fenotipos, y sus representaciones dentro del algoritmo evolutivo, que llamamos genotipos (Morrison, 1998). Esto implica definir cómo cada fenotipo se mapea a un conjunto específico de genotipos que los representan. Por ejemplo, se pueden representar números enteros utilizando su equivalente en código binario, donde el número 10 se representaría como 1010 en su correspondiente genotipo.

Se debe tener en cuenta que la búsqueda evolutiva tiene lugar en el espacio genotípico. La terminología que se suele utilizar dentro de los algoritmos evolutivos usa sinónimos para nombrar los elementos de los dos espacios (fenotípico y genotípico). Para los elementos del conjunto de fenotipos, se utilizan los términos solución candidata, fenotipo e individuo. En cuanto a los elementos del conjunto de genotipos, se puede utilizar los términos genotipo, cromosoma e individuo.

La representación puede hacer referencia al mapeo del espacio fenotípico al espacio genotípico, pero también se puede usar en un sentido ligeramente diferente, donde el énfasis está en la estructura de datos del espacio genotípico, interpretación que suele encontrarse, por ejemplo, al hablar sobre operadores de mutación para la representación binaria.

Una de las representaciones más básicas que puede utilizarse es una cadena binaria, como se puede ver en la [Figura 2.1](#)

1	0	1	0	0	0	1	1
---	---	---	---	---	---	---	---

Figura 2.1: Representacion de un individuo

Es importante notar que a veces el espacio de genotipos puede exceder al de fenotipos. Por ejemplo, si se utiliza un código binario para representar números hasta el 10 (1010 en binario), el espacio genotípico puede incluir representaciones adicionales que van más allá de 1010, llegando hasta 1111 (15 en decimal). Sin embargo, estos genotipos adicionales (del 11 al 15) no se utilizan en la solución

del problema original y hay que tenerlos en cuenta a la hora de definir individuos válidos. Una de las estrategias comunes para manejar el problema del mapeo de genotipos a fenotipos suele ser, en lugar de descartar genotipos adicionales, descartar a los individuos que no pretenezcan al espacio de fenotipos, de manera que no sean seleccionados para su reproducción.

### **Función de evaluación o fitness**

La función de evaluación, también conocida como función de fitness, es un componente esencial en un algoritmo evolutivo. Su objetivo es determinar qué tan buena es cada individuo en términos del problema que se está intentando resolver. La función asigna un valor de fitness a cada individuo, lo que permite cuantificar su calidad (Eiben y Smith, 2015). En el caso de que el algoritmo busque maximizar el valor de la función, un individuo con un valor de fitness más alto representa una mejor solución para el problema dado. De lo contrario, un valor de fitness más bajo representará una peor solución. La función de evaluación guía el proceso evolutivo, ya que los individuos con mayor fitness tienen mayores probabilidades de ser seleccionados para reproducción, lo que incrementa la eficacia del algoritmo para encontrar soluciones óptimas. La función de fitness, además de definirse como un procedimiento que asigna una media de calidad a los genotipos, también se puede considerar como una función que representa los requisitos a los que adaptarse, definiendo lo que significa mejorar y que representa la tarea a resolver dentro del contexto evolutivo.

### **Población**

La población en un algoritmo evolutivo es el conjunto de individuos que conforman una generación dentro del proceso evolutivo. Los individuos por sí solos no pueden cambiar ni adaptarse, es la población la que lo hace, por lo que se dice que constituye la unidad de evolución. Inicialmente, la población se crea aleatoriamente o mediante estrategias específicas y evoluciona a lo largo del tiempo mediante procesos de selección, cruce y mutación. La diversidad de la población es crucial, ya que una mayor diversidad genética permite explorar una gama más amplia de posibles soluciones y evita la convergencia prematura hacia óptimos locales. La población se actualiza en cada iteración del algoritmo, buscando mejorar el valor de fitness de la población en general a través de sucesivas generaciones.

En la mayoría de las aplicaciones de algoritmos evolutivos, el tamaño de la población es constante durante la búsqueda evolutiva.

### **Selección**

La selección es el proceso mediante el cual se eligen individuos de la población actual para reproducirse y formar la siguiente generación. El proceso de selección está influenciado por el fitness de los individuos: aquellos con mejor fitness tienen más probabilidades de ser seleccionados para ser padres de la siguiente

generación. Existen varios métodos de selección, como la selección por torneo, la ruleta de selección y la selección por rango, cada uno con sus propias ventajas y desventajas. La selección es crucial para asegurar que las características de los individuos más aptos se propaguen a través de las generaciones, mejorando así la calidad global de la población.

A diferencia de los operadores de variación, que actúan sobre uno o dos individuos, los operadores de selección (selección de padres y selección de supervivientes) operan a nivel de población.

### **Operadores de variación**

El papel de los operadores de variación es crear nuevos individuos a partir de los existentes. Los operadores de variación se dividen en dos tipos según su aridad, siendo los operadores binarios considerados de cruzamiento y los unarios llamados operadores de mutación.

### **Recombinación o cruce**

El cruce o recombinación, implica la combinación de material genético de dos individuos seleccionados para formar nuevos individuos, denominados descendientes. El proceso de cruce simula la recombinación genética biológica y permite explorar nuevas soluciones en el espacio de búsqueda. Durante el cruce, segmentos de los cromosomas de los padres se intercambian para crear descendientes que contienen una mezcla de características de ambos progenitores. La recombinación es esencial para la exploración del espacio de soluciones y para la introducción de variabilidad en la población, lo que ayuda a evitar el estancamiento en óptimos locales.

El operador de cruce fusiona información de dos genotipos en uno o dos genotipos descendientes. La elección de qué partes de cada progenitor se combinan, y la forma en la que se combinan, dependen del operador de cruce que se utiliza. El principio detrás de la recombinación es simple: al cruzar dos individuos con características diferentes pero deseables, se puede producir un descendiente que combine las características de ambos padres. Los algoritmos evolutivos crean una serie de descendientes aceptando que algunos tendrán combinaciones indeseables de rasgos, la mayoría puede no ser mejor ni peor que sus padres, y esperando que algunos tengan características mejoradas. Los operadores de recombinación en los algoritmos evolutivos generalmente se aplican de manera probabilística, con una probabilidad existente de no ser aplicados.

### **Mutación**

La mutación es considerada un operador de variación unario. Introduce variabilidad en la población modificando aleatoriamente los individuos seleccionados, cambiando uno o más genes en un cromosoma y alterando así el fenotipo del individuo. La mutación es crucial para evitar la convergencia prematura hacia óptimos locales y permitir una exploración más amplia del espacio de búsqueda.

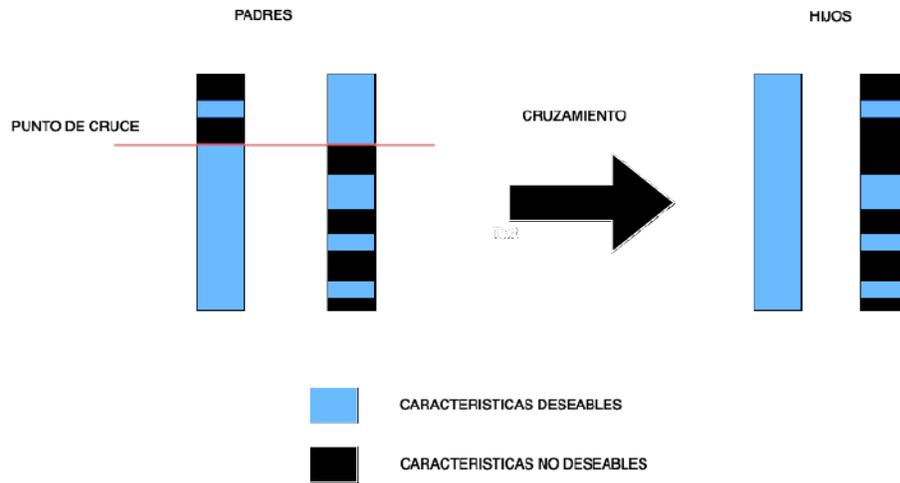


Figura 2.2: Cruce de individuos

Aunque la mutación ocurre generalmente con baja probabilidad, su impacto es significativo ya que puede introducir nuevas características en la población y ayudar a descubrir soluciones que no habrían sido posibles solo con el cruce.

Es importante tener en cuenta que la selección de los operadores de variación a utilizar pueden depender de la representación. Para diferentes representaciones se deben definir diferentes operadores de variación. Por ejemplo, si los genotipos son cadenas de bits, entonces invertir un 0 en un 1 (o un 1 en un 0) puede usarse como operador de mutación. Sin embargo, si representamos a los individuos mediante estructuras similares a árboles, se requiere otro operador de mutación.

#### 2.1.4. Algoritmos evolutivos multi-objetivo

Un problema de optimización multi-objetivo involucra la optimización de un conjunto de funciones, ya sea minimizándolas o maximizándolas. La existencia de múltiples funciones objetivo plantea una diferencia fundamental con un problema mono-objetivo: no existirá una única solución al problema, sino un conjunto de soluciones que plantean diferentes compromisos entre los valores de las funciones a optimizar. Desde el punto de vista multi-objetivo no existe una definición de óptimo global, como pudiera ser fácilmente entendida en el caso de la optimización con un solo objetivo ([Zitzler y cols., 2000](#)).

El hecho de que las soluciones a los problemas multi-objetivo representen compromisos entre distintos objetivos significa que mejorar el rendimiento en un objetivo puede implicar empeorar el rendimiento en otro. En optimización mono-objetivo se busca un vector de decisión  $n$ -dimensional que optimice una función escalar, mientras que en optimización multi-objetivo se intenta encontrar uno que optimice una función vectorial cuyos elementos representan las distin-

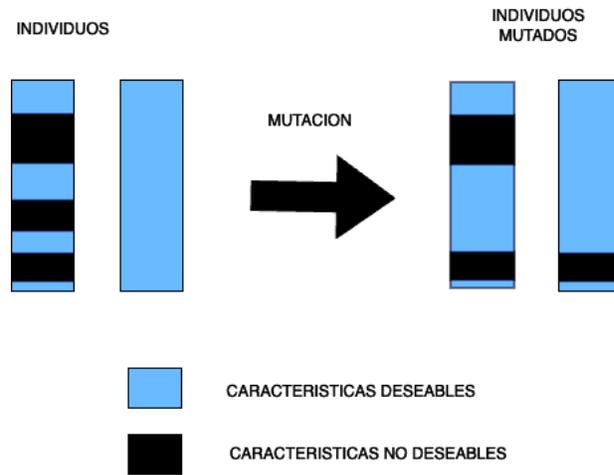


Figura 2.3: Mutacion de individuos

tas funciones objetivo. Cuando existen objetivos contradictorios, el espacio de búsqueda no está completamente ordenado, lo que significa que dos soluciones pueden ser igualmente preferibles entre sí. Es poco común que una sola variable de decisión optimice todos los objetivos simultáneamente.

En los problemas multi-objetivo se acostumbra a considerar que sus soluciones están formadas por todas las soluciones factibles no dominadas. Una solución factible es no dominada cuando las componentes de la función objetivo no pueden ser igualadas por ninguna otra solución factible y al menos una de ellas no puede ser mejorada estrictamente por ninguna otra solución factible. Estas soluciones se llaman Pareto óptimas o eficientes. El enfoque multi-objetivo proporciona un conjunto de soluciones, donde ninguna solución es absolutamente mejor que otra sin empeorar al menos uno de los objetivos, conocidas como el frente de Pareto. Esto permite a los tomadores de decisiones elegir la solución que mejor se adapte a sus prioridades y restricciones específicas. Además, facilita el análisis de los trade-offs entre diferentes objetivos, crucial en situaciones donde se deben equilibrar diferentes aspectos.

## NSGA-II

El Algoritmo Genético No Dominado Ordenado (NSGA-II por sus siglas en inglés) es uno de los algoritmos evolutivos multi-objetivos más populares y efectivos y ha destacado por su eficiencia y capacidad para mantener la diversidad de soluciones (Yu y Gen, 2010). Como su nombre indica, está basado en la observación de la dominancia de las soluciones y con características avanzadas para preservar la diversidad. Es un algoritmo utilizado para resolver problemas multi-objetivo.

Las características principales de NSGA-II incluyen el uso de un ordenamiento no dominado elitista, facilitado por una técnica de comparación que emplea una población auxiliar y reduce así la complejidad de los chequeos de dominancia. Además, se preserva la diversidad mediante una técnica de crowding que no requiere la especificación de parámetros adicionales. Los valores de aptitud se asignan en función de rangos de no dominancia.

La técnica de clasificación por crowding utilizada en NSGA-II consiste en ordenar los puntos del último frente permitido en orden descendente según su valor de distancia de crowding y seleccionar los primeros de la lista que quepan en la población de la siguiente generación. La distancia de crowding de un punto es la medida del espacio alrededor de él que no está ocupado por ningún otro individuo de la población. Para calcular la distancia de crowding, se estima el perímetro del cuboide creado con los puntos más cercanos como vértices.

Los algoritmos evolutivos para optimización multi-objetivo buscan cumplir dos propósitos simultáneamente: obtener un conjunto de soluciones cercanas al verdadero frente de Pareto o a su aproximación (convergencia) y mantener la diversidad del frente de soluciones. En NSGA-II, la convergencia se logra a través del proceso evolutivo, mientras que la diversidad se mantiene mediante técnicas específicas como crowding.

En cada generación  $i$ , la población de descendientes  $Q_i$  se crea a partir de la población padre  $P_i$  utilizando operadores habituales. Ambas poblaciones se combinan para formar  $R_i$ , que incluye a todos los individuos de ambas. Posteriormente, mediante un ordenamiento no dominado, se clasifica  $R_i$  en diferentes frentes de Pareto. Luego de determinar las clases de no dominancia ( $F_1, F_2, \dots$ ), se agregan incrementalmente a la población de la siguiente generación las soluciones de cada clase de dominancia obtenida, comenzando por las mejores y continuando hacia las menos aptas. Dado que  $R_i$  tiene el doble del tamaño de la población utilizada por el algoritmo, se descartan las soluciones de las clases de dominancia menos aptas. Al agregar los puntos del último frente permitido, puede suceder que no todas las soluciones quepan en los espacios restantes de la siguiente población. En este caso, se eligen los puntos que maximizan la diversidad mediante la técnica de clasificación por crowding. (Deb y cols., 2002)

## 2.2. Redes generativas antagónicas

Las Redes Generativas Antagónicas (GANs por sus siglas en inglés) son un tipo de modelo generativo en el campo del aprendizaje automático y la inteligencia artificial. Las GANs constan de dos redes neuronales principales, conocidas como el generador y el discriminador, que se entrenan en conjunto en un proceso competitivo (Wang y cols., 2017).

El generador busca capturar la distribución subyacente de las muestras reales y produce nuevas muestras de datos. Toma como entrada un vector de ruido aleatorio (o datos reales) y lo mapea a una nueva muestra de datos, como una imagen. La meta del generador es aprender a generar datos que sean indistinguibles de los datos reales observados.

El discriminador actúa como un clasificador binario que intenta distinguir entre datos reales (tomados del conjunto de datos de entrenamiento) y datos generados por el generador. El discriminador aprende a asignar probabilidades a las muestras, indicando si cada muestra es real o generada.

El generador y el discriminador pueden adoptar la estructura de las redes neuronales profundas. La optimización en las GANs se estructura como un juego de minmax, con el objetivo de lograr un punto de equilibrio en el cual se considere que el generador ha aprendido la distribución de las muestras reales. El generador y el discriminador se entrenan de manera adversarial, el generador intenta engañar al discriminador generando muestras que el discriminador clasificaría como reales, mientras que el discriminador mejora su capacidad para distinguir entre datos reales y generados. Durante el entrenamiento, el generador y el discriminador se actualizan de manera alternada para mejorar sus respectivas habilidades. El entrenamiento busca llegar a una convergencia donde el generador es capaz de generar muestras que son indistinguibles de las muestras reales para el discriminador.

Las GANs pueden modelar distribuciones de probabilidad complejas y arbitrarias, lo que las hace adecuadas para tareas de generación de datos realistas como imágenes y sonido. Han sido particularmente exitosas en la generación de imágenes realistas, superando métodos tradicionales en términos de calidad visual. Sin embargo, las GANs sufren de muchos problemas, especialmente durante el entrenamiento. Un modo común de fallo implica que el generador colapse y produzca sólo una muestra o una pequeña familia de muestras muy similares. Otro posible problema es que el generador y el discriminador oscilen durante el entrenamiento, en lugar de converger a un punto fijo. Además, si un agente se vuelve mucho mejor en su tarea en comparación al otro, la señal de aprendizaje para su contraparte se vuelve inútil y el sistema no aprende.

Los modelos obtenidos mediante el aprendizaje con métodos generativos, juegan un papel esencial en el campo de la inteligencia artificial. Los métodos generativos involucran hipótesis de distribución y estimación de parámetros, y pueden muestrear nuevos datos a partir de los modelos estimados.

## StyleGAN

StyleGAN (Generative Adversarial Network for Style Transfer) es una arquitectura de red generativa antagónica (GAN) desarrollada por NVIDIA, que se ha destacado por su capacidad para generar imágenes de alta calidad y resolución (Karras y cols., 2018). Es particularmente conocida por su capacidad para producir imágenes de rostros humanos que parecen increíblemente realistas. StyleGAN se basa en las ideas de transferencia de estilo y ha introducido innovaciones que permiten un control más fino sobre las características generadas en las imágenes.

StyleGAN propone una arquitectura alternativa para el generador en redes generativas adversariales (GAN), inspirada en la literatura de transferencia de estilo, donde el generador expone nuevas formas de controlar el proceso de síntesis de imágenes (Metz y cols., 2017). El generador comienza con una entrada

constante aprendida y ajusta el estilo de la imagen en cada capa de convolución basándose en el código latente, controlando así directamente la intensidad de las características de la imagen en diferentes escalas. Combinado con el ruido inyectado directamente en la red, el cambio arquitectónico permite una separación automática y no supervisada de atributos de alto nivel (como la pose y la identidad) de la variación estocástica (como pecas y cabello) en las imágenes generadas. Además, facilita operaciones intuitivas de mezcla e interpolación específicas para diferentes escalas.

El generador de StyleGAN utiliza un código latente que se transforma en un espacio latente intermedio. El espacio no está restringido a seguir la distribución de probabilidad de los datos de entrenamiento, permitiendo una representación más desentrelazada de los factores de variación. Al utilizar el código latente de entrada en el espacio intermedio, se logra una representación más lineal y menos entrelazada de los diferentes factores de variación.

En resumen, el generador basado en estilos de StyleGAN introduce un cambio significativo en la forma en que se generan y controlan las imágenes en las GAN, ofreciendo mejoras sustanciales en calidad y control sobre los generadores tradicionales.

## DeepFace

DeepFace es una biblioteca ligera de reconocimiento facial y análisis de atributos faciales para Python. Envuelve modelos de análisis facial, como VGG-Face, FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib SFace, así como varios modelos de detección facial como OpenCV, Ssd, Dlib, MtCnn, Faster MtCnn y RetinaFace. También es capaz de predecir edad, género, expresión facial (como enojo, miedo, neutral, triste, disgusto, felicidad y sorpresa) y raza (asiático, blanco, medio oriental, indio, latino y negro).

Facebook DeepFace (FDF) es un sistema avanzado de análisis facial. Es uno de los modelos pioneros en utilizar redes neuronales profundas para identificar y verificar rostros humanos con alta precisión. Representa imágenes faciales como vectores multidimensionales. La función de representación devuelve una lista de embeddings con diferentes dimensiones según el modelo utilizado.

FDF emplea una red neuronal profunda con múltiples capas para extraer características faciales de las imágenes. La arquitectura es especialmente eficaz para procesar datos visuales. El modelo fue entrenado utilizando un gran conjunto de datos de imágenes faciales etiquetadas, provenientes de la vasta colección de fotos de usuarios de Facebook. Esto permitió al modelo aprender una amplia variedad de características faciales y mejorar su precisión. FDF alcanzó una precisión de reconocimiento facial superior al 97% (Taigman y cols., 2014).

## Funcionamiento de DeepFace

Se describen las etapas fundamentales para realizar el reconocimiento facial, según el marco de trabajo de DeepFace.

- **Detectar:** Etapa donde se busca encontrar el o los rostros en la imagen de entrada.
- **Alinear:** El objetivo de la alineación es generar una cara con orientación frontal a partir de la imagen de entrada. La imagen facial se preprocesa para corregir la orientación y la posición. Utilizando un modelo 3D, se alinea el rostro a una vista frontalizada, estandarizando la posición y el ángulo del rostro. Esto reduce la variabilidad causada por diferentes poses y ángulos de la cabeza.
- **Representar:** Consiste en abstraer las características y la información de los rasgos del rostro en un vector de dimensiones fijas, también llamado vector de embeddings. Se ha desarrollado un modelo que aprende a extraer características directamente de los datos. Al ser entrenado con grandes volúmenes de datos faciales, el modelo aprende a identificar y abstraer automáticamente características faciales relevantes.
- **Clasificar:** La etapa de clasificación verifica si dos imágenes de rostros pertenecen a la misma persona o a personas diferentes. Los vectores de embeddings obtenidos en la etapa de representación de los dos rostros se comparan utilizando una métrica de similitud, comúnmente la distancia euclidiana. Si la distancia entre los vectores es menor que un umbral predefinido, las imágenes se consideran como pertenecientes a la misma persona. Si la distancia es mayor, se consideran como personas diferentes.

DeepFace ha demostrado ser una herramienta poderosa para aplicaciones de reconocimiento facial, aunque también plantea importantes consideraciones éticas y de privacidad, incluyendo la necesidad de obtener consentimiento informado, la transparencia en el uso de datos y la minimización de sesgos para evitar discriminación. Además, se deben implementar regulaciones y medidas de seguridad para proteger la privacidad personal y prevenir el abuso de la tecnología.

## RetinaFace

RetinaFace es un detector facial avanzado basado en aprendizaje profundo para Python, que incluye la detección de puntos de referencia faciales. Es uno de los detectores de rostro que DeepFace utiliza en su pipeline de reconocimiento facial. Es conocido por su alto rendimiento en la detección y alineación de rostros, lo que contribuye significativamente a mejorar la precisión del reconocimiento facial.

RetinaFace es muy eficaz en detectar rostros incluso en imágenes con múltiples personas o en condiciones difíciles. Además, realiza una detección precisa de puntos de referencia faciales (ojos, nariz, boca), lo que mejora la alineación de los rostros. Permite que todos los procesos de detección, alineación y normalización, mejoren la precisión global del reconocimiento facial.

Aunque RetinaFace es más preciso que otros detectores como OpenCV o SSD, también es más lento. Esto hace que sea más adecuado para aplicaciones donde la precisión es más importante que la velocidad.

## Capítulo 3

# Revisión de antecedentes

En el presente capítulo se detalla la investigación realizada sobre el estado del arte en conceptos como GANs, el uso de algoritmos evolutivos para exploración de espacios latentes, así como algoritmos para la generación de rostros humanos.

### 3.1. Motivación y metodología

El propósito de la revisión de antecedentes fue explorar el estado del arte de la exploración del espacio latente de las redes neuronales generativas antagónicas a través del uso de algoritmos evolutivos, con el objetivo final de desarrollar un algoritmo generativo multi-objetivo para la generación de rostros artificiales realistas con determinados atributos de raza, género y edad.

Para llevar a cabo la investigación se buscaron artículos académicos abarcando tres categorías. La primera buscó obtener un panorama general del uso de redes neuronales generativas antagónicas (GANs) en el ámbito académico. La segunda categoría se enfocó en el uso de algoritmos evolutivos para la exploración del espacio latente de redes neuronales generativas antagónicas sin limitarse a la generación de rostros. Finalmente se investigaron artículos de generación de rostros a través del uso de algoritmos evolutivos para la exploración del espacio latente de redes neuronales generativas antagónicas.

### 3.2. Uso de GANs y algoritmos evolutivos para explorar el espacio latente de GANs

En esta sección, como introducción al uso de GANs en el ámbito académico, se releva el artículo de [Pitiddu y Veiga \(2022\)](#), en el que compararon diversos modelos de clasificadores basados en GANs para la identificación de dígitos manuscritos generados también por GANs. Dentro de la categoría de exploración del espacio latente de redes neuronales generativas antagónicas se destacan los artículos presentados por [Roziere y cols. \(2020\)](#), que expuso una introducción

al uso de optimización evolutiva para mejorar los vectores utilizados en GANs, [Fernandes y cols. \(2020\)](#), que presentó una prueba de concepto de la posibilidad de guiar la exploración del espacio latente de GANs a través de distintos algoritmos evolutivos, y [Galatolo y cols. \(2021\)](#), donde se desarrolló un framework llamado CLIP-GLaSS con el objetivo de generar texto a partir de imágenes y viceversa.

## **Diseño de clasificadores de dígitos manuscritos generados por redes GAN**

[Pitiddu y Veiga \(2022\)](#) emplearon imágenes de dígitos manuscritos generados por dos redes denoising-GAN: una con capas de convolución y otra con capas densas. El objetivo fue clasificar estas imágenes mediante un proceso de encuesta subjetiva, seguido de la automatización de la clasificación a través de redes neuronales y otros algoritmos. En las encuestas, se presentaron grupos de 40 imágenes de dígitos generados por las GANs a un conjunto de personas, quienes debían determinar si consideraban que cada imagen era legible o no. Se realizaron dos encuestas en total, con la participación de 94 personas. Tras analizar los resultados de las encuestas, se propusieron varios clasificadores basados en redes neuronales para imitar el proceso de clasificación humano.

Los clasificadores propuestos incluyeron un clasificador de umbrales, un Discriminador Perceptrón Multicapa (MLPD), clasificadores con 10 y 20 entradas y por último clasificadores ponderados, donde se combinaron las salidas de diferentes clasificadores para obtener una clasificación final. Se llevaron a cabo pruebas de entrenamiento y validación con distintos conjuntos de datos, además de explorar enfoques para ampliar el conjunto de datos, como el uso de imágenes trasladadas o GANs. Con base en los resultados de la experimentación, se concluyó que el clasificador de umbrales requería un proceso heurístico manual y más tiempo de ejecución. El clasificador MLPD se basaba en una única red neuronal, mientras que los clasificadores de 10 y 20 entradas requerían entrenar dos redes (una del numerador 0-9 y otra propia). El clasificador ponderado implicaba entrenar tres redes. Las pruebas indicaron que el clasificador de 10 entradas con función sigmoide era el que mejor representaba la opinión mayoritaria en la encuesta. Por lo tanto, se seleccionó el clasificador para imitar el comportamiento promedio de una persona en la clasificación de imágenes de números manuscritos.

El artículo presenta una base teórica y práctica sobre la aplicación de GANs en la generación de imágenes sintéticas y el análisis de las mismas. Su utilidad en el proyecto planteado consiste en el conocimiento de la utilidad de las redes neuronales en el ámbito académico.

## EvoGAN: Evolutionary Generative Adversarial Networks

Roziere y cols. (2020) utilizaron una optimización evolutiva para explorar el espacio latente de las GANs, empleando estimadores de calidad para guiar la búsqueda. Los resultados demostraron que la metodología producía imágenes de mayor calidad mientras se mantenía la diversidad de la GAN original. Además, se estableció que la técnica podía reemplazar fácilmente a las GANs convencionales siempre que hubiera un estimador de calidad disponible para las salidas de la GAN.

El método desarrollado, denominado EvoGAN, se diseñó para mejorar la calidad de las imágenes generadas por redes generativas adversarias (GANs) sin alterar su proceso de entrenamiento. EvoGAN utilizó un generador  $G$  creado por una GAN, el cual mapeaba un vector latente  $z$  a una imagen  $G(z)$ . En lugar de seleccionar  $z$  de manera aleatoria, EvoGAN optimizó  $z$  para maximizar un criterio de calidad  $Q$ , utilizando una estrategia de optimización evolutiva. El algoritmo de optimización empleado fue una variante de la estrategia de evolución  $(1 + 1)$  con tasas de mutación aleatorias, adaptadas para mantener la diversidad y mejorar la calidad de las imágenes generadas. El algoritmo EvoGAN equilibró la calidad y la diversidad mediante la manipulación de la tasa de mutación  $\alpha$ .

Se aplicó el método EvoGAN en tres modelos de redes generativas adversarias (GAN): (i) StyleGAN2 para rostros, gatos, caballos y obras de arte; (ii) PokeGAN para montañas y Pokémon; y (iii) PGAN del repositorio Pytorch GAN Zoo para FashionGen.

Se evaluó el rendimiento de EvoGAN en comparación con StyleGAN mediante un estudio humano, donde los evaluadores compararon generaciones en pares y seleccionaron sus preferencias. Los resultados revelaron que, en general, EvoGAN mejoraba la calidad de las imágenes generadas por StyleGAN2. En resumen, los experimentos demostraron que EvoGAN podía mejorar la calidad de las imágenes generadas por diversas GANs en distintos conjuntos de datos, al mismo tiempo que mantenía la diversidad y la consistencia. El método se basó en optimizar el vector latente de entrada del generador  $G$  de la GAN utilizando un algoritmo evolutivo. Se demostró que al optimizar el vector con un valor  $\alpha$  pequeño y utilizando el criterio de calidad Koncept512, se lograban mejoras significativas en la calidad y diversidad de las imágenes generadas en diversos casos, incluyendo GANs para caras, animales, arte y moda. En comparación con la búsqueda aleatoria, se observó que los algoritmos evolutivos eran más efectivos en mejorar la calidad de las imágenes generadas, y se constató que, en la mayoría de los casos, se mantenía la diversidad de las imágenes al utilizar un  $\alpha$  pequeño.

El artículo de Roziere y cols. (2020) representó en el contexto de el proyecto una clara introducción al uso de la optimización evolutiva para la generación de imágenes en GANs. Mostró que incluso el uso de algoritmos evolutivos extremadamente sencillos como  $1+1$  generan cambios significativos en la generación de imágenes. Introdujo además el uso de StyleGAN2 para la generación de imágenes.

## Evolutionary latent space exploration of generative adversarial networks

[Fernandes y cols. \(2020\)](#) buscaron soluciones en el espacio latente siguiendo criterios predefinidos. La idea principal fue habilitar la búsqueda y generación de conjuntos de vectores latentes que cumplieran un objetivo específico. Se utilizaron algoritmos genéticos (GAs) y MAP-Elites para asistir en la tarea.

Para explorar el espacio latente, se generaron conjuntos de individuos utilizando enfoques como el Muestreo Aleatorio (RS), Algoritmos Genéticos (GA) y MAP-Elites. Los tres enfoques buscaron evolucionar conjuntos de individuos, pero variaron en la forma en que promovieron el cambio a lo largo de las iteraciones. En el enfoque de Muestreo Aleatorio (RS), se creó un conjunto completamente nuevo de individuos al comienzo de cada generación y se evaluó mediante una función de evaluación. GA partió de una población inicial aleatoria y utilizó operadores de variación (mutación y cruzamiento) para evolucionar la población. MAP-Elites mantuvo un mapa de combinaciones plausibles de características y creó nuevos individuos a partir de operadores de variación.

Los individuos se inicializaron como conjuntos de imágenes y se evaluaron utilizando una función de similitud promedio. Se aplicaron operadores de cruzamiento y mutación, y se utilizaron métricas de similitud como el Error Cuadrático Medio (RMSE) y la Correlación Cruzada Normalizada (NCC). Los experimentos realizados en el dominio de imágenes utilizando diferentes conjuntos de datos demostraron la viabilidad de aplicar el enfoque propuesto a distintos tipos de GANs, abarcando desde imágenes en escala de grises hasta imágenes a color. Los resultados de estos experimentos sirvieron como una prueba de concepto, confirmando que era efectivamente posible guiar la generación de variables latentes hacia criterios específicos.

El artículo de [Fernandes y cols. \(2020\)](#) presenta una prueba de concepto sobre el uso de variados algoritmos evolutivos con distinta complejidad para la exploración del espacio latente en la generación de imágenes. Además, se introduce el uso de algoritmos genéticos, con el uso de operadores no solo de mutación sino de cruzamiento en la evolución de la población.

## Generating images from caption and viceversa via clip-guided generative latent space search

[Galatolo y cols. \(2021\)](#) desarrollaron un framework llamado CLIP-GLaSS con el objetivo de generar texto a partir de imágenes y viceversa. El proceso se basó en el uso de una GAN y un algoritmo evolutivo para explorar el espacio latente y generar resultados de alta calidad.

El framework constó de tres componentes principales:

- La red CLIP, encargada de crear embeddings para imágenes y textos. Estos embeddings son similares si los conceptos que representan son similares.

- Un generador de imágenes o texto para producir resultados con embeddings similares. Para generar imágenes, se utilizaron BigGAN y StyleGAN2, mientras que para la generación de texto se empleó GPT-2.
- Un Autoencoder (AE) destinado a explorar el espacio latente del generador para encontrar embeddings con alta similitud entre la imagen y el texto. Se utilizó el algoritmo NSGA-II para problemas multi-objetivo y un algoritmo genético clásico para problemas mono-objetivo.

En el caso de la generación de texto a partir de imágenes, el AE tuvo dos funciones objetivo. La primera maximizaba la similitud coseno entre los embeddings de la imagen generada por el generador y el embedding del texto objetivo. La segunda minimizaba la pérdida del discriminador, calculada usando la codificación generada a partir de la imagen generada y el valor asociado a una imagen real. Para la generación de imágenes a partir de texto, el problema se abordó como un problema mono-objetivo debido a la falta de un Discriminador. La función de fitness se basó en la distancia entre el embedding del texto generado y el embedding de la imagen original.

La calidad de los resultados se evaluó mediante una revisión manual de las imágenes y textos generados, clasificándolos en categorías de baja, media y alta relevancia. También se realizaron pruebas de optimización sin el uso del discriminador.

Se concluyó que las imágenes generadas eran relevantes en comparación con el texto de entrada, aunque podría mejorarse si el generador se entrenara con una mayor variedad de imágenes. Además, se determinó que el papel del discriminador tiene una influencia significativa en la calidad general de los resultados.

La significancia de el artículo en el proyecto es el uso de embeddings en la representación de imágenes, concepto que se repite en los artículos de generación de rostros y es utilizado también en la solución presentada en el proyecto. Se introduce además la búsqueda de soluciones con múltiples objetivos y el uso del algoritmo NSGA-II para la optimización.

### **3.3. Generación de rostros a través del uso de algoritmos evolutivos para la exploración del espacio latente de GANs**

La categoría de búsqueda de generación de rostros con algoritmos evolutivos y redes neuronales generativas antagónicas es la más aplicada al desarrollo del proyecto planteado. Se presentan en esta sección los artículos estudiados dentro de esta categoría.

## Evolution of images with diversity and constraints using a generative adversarial network

Neumann y cols. (2018) utilizaron GANs para generar imágenes aplicando una búsqueda evolutiva en el espacio latente con el fin de maximizar y minimizar características de una y dos dimensiones en imágenes de rostros y de mariposas. Se buscó la optimización del realismo de la imagen y de la característica buscada. Las características fueron: hue, mean-saturation, smoothness, reflectional symmetry y global contrast factor.

Se experimentó con (1+1) EA y Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES), dando la segunda mejores resultados, en los valores de la función objetivo y en tiempo de ejecución. Se utilizó el valor de las características (utilizando las métricas que definen a cada una) y se multiplicó además por el valor retornado por el discriminador, ya que de lo contrario el generador fallaba en producir imágenes realistas. Se experimentó también con una función de cut-off que permitió sacrificar realismo de la imagen en pos de la obtención de mejores resultados en las características buscadas.

Se llevaron a cabo experimentos de una sola dimensión utilizando conjuntos de datos de rostros y mariposas, optimizando características como tono, saturación, suavidad, simetría y GCF. Se observó que la característica de tono tiene el rango más amplio en cuanto a valores de características. Las imágenes generadas con el conjunto de datos de rostros parecen más reales que las generadas con el conjunto de datos de mariposas. Las imágenes generadas a partir del conjunto de datos de mariposas logran un rango de características más alto.

En el siguiente experimento, se evolucionaron imágenes utilizando GAN para minimizar y maximizar dos dimensiones de características. Estos experimentos buscaron comprender la interacción entre las características y el impacto de las prioridades de imagen incorporadas en el GAN en la optimización de características. Se realizaron experimentos con combinaciones de características como GCF-saturación, GCF-suavidad, tono-saturación, tono-simetría, saturación-simetría y suavidad-saturación. Los resultados mostraron que debido a conflictos inherentes, como que un alto GCF requiere alto contraste y que la suavidad requiere bajo contraste, ciertas combinaciones de características son difíciles de optimizar simultáneamente. Además, se observó en particular que las imágenes generadas de mariposas presentan una mayor variabilidad en las características de saturación y simetría, mientras que las caras tienden a ser más realistas cuando se maximizan ciertas características específicas.

El artículo propone el uso de algoritmos evolutivos más avanzados, como CMA-ES, y la combinación de características objetivo para optimizar, lo que lo hace más relevante para nuestro proyecto en comparación con los estudios anteriores. Además se introduce la generación de rostros dentro de las imágenes a optimizar.

## **Evolutionary latent space search for driving human portrait generation**

**Machín y cols. (2021)** propusieron encontrar vectores en el espacio latente de redes neuronales generativas con el objetivo de que las imágenes generadas fueran similares a imágenes objetivo. Debido al amplio tamaño del espacio latente, se utilizaron algoritmos evolutivos con esta finalidad. Se utilizó FaceNet para evaluar la similitud entre los rostros generados y los objetivos, mientras que el modelo generativo empleado fue StyleGAN2. Se utilizó un algoritmo mono-objetivo en el que la codificación de la solución consistió en un vector de 512 números de punto flotante, que representaban las componentes de los vectores en el espacio latente de StyleGAN2. La función de fitness se definía como el opuesto de la distancia euclidiana entre los vectores generados por FaceNet para la imagen objetivo y la generada.

Se optó por un operador de inicialización aleatoria siguiendo una distribución normal. La selección se basó en torneos, empleando el método de "blend-alpha crossover" para el cruzamiento. Las mutaciones se realizaron mediante una distribución gaussiana. Después de implementar el algoritmo y optimizarlo mediante el cálculo de fitness en lotes (batches), se lograron generar imágenes capaces de engañar a FaceNet en menos de 15 minutos. Esto demostró que explorar el espacio latente a través de algoritmos evolutivos representaba una alternativa al entrenamiento de redes neuronales, evitando la necesidad de una capacidad de cómputo considerable.

En términos de similitud, ambos resultados presentaron una mejora en la comparación entre la imagen original y otras imágenes diferentes de la misma persona. En cuanto a la diversidad de las imágenes generadas, también se obtuvieron resultados satisfactorios.

## **Multi-target evolutionary latent space search of a generative adversarial network for human face generation**

**Machín y cols. (2022)** presentaron un enfoque evolutivo para la generación de imágenes sintéticas de rostros humanos con múltiples objetivos, basado en la exploración del espacio latente de las redes generativas adversarias. El enfoque propuesto buscó generar imágenes de rostros humanos distintos pero con similitudes a dos imágenes objetivo predefinidas. La optimización utilizó redes generativas adversarias para la generación de rostros, reconocimiento facial para evaluar la similitud y un algoritmo evolutivo para explorar el espacio de búsqueda.

La propuesta se basó en StyleGAN2 para generar imágenes de rostros humanos que cumplieran ciertos criterios, como la similitud con las imágenes objetivo. La búsqueda de un espacio latente adecuado para generar imágenes que cumplan con estos criterios se realizó mediante un enfoque evolutivo que evaluó la similitud entre las imágenes generadas y las imágenes objetivo. El modelo utilizado para evaluar la calidad de las imágenes fue FaceNet.

El enfoque evolutivo propuesto empleó una representación de solución que codificaba vectores en el espacio latente de StyleGAN2 como individuos compuestos por 512 números de punto flotante. La función de aptitud evaluó la similitud de cada imagen facial generada con los rostros objetivo. La similitud se evaluó mediante la distancia coseno entre los embeddings de dos imágenes. La función de aptitud utilizó una combinación ponderada lineal de las distancias a los dos rostros objetivo. Para iniciar la búsqueda en el espacio latente, se generó la población de soluciones mediante una distribución normal (gaussiana) con media cero y desviación estándar uno. La inicialización aleatoria proporcionó valores útiles para dar inicio a la búsqueda sin necesitar información precisa sobre las características del espacio latente.

Después de la evaluación inicial, se optó por el operador de selección por torneo. Los resultados más destacados se lograron con un tamaño de torneo de tres individuos, de los cuales solo uno sobrevivió como progenitor. Se utilizó el operador de recombinación BLX- $\alpha$  (Blend- $\alpha$ ), que selecciona valores uniformemente entre dos puntos (en el espacio de codificación) que incluyen a los dos progenitores. El parámetro  $\alpha$  determinó la extensión equitativa en ambas direcciones desde los puntos seleccionados.

El operador que generó los mejores resultados fue la mutación gaussiana, utilizando parámetros medios  $\mu = 0$  y desviación estándar  $\sigma = 1$ . El análisis empírico demostró que la configuración  $\mu = 0$  y  $\sigma = 1$  pudo generar la diversidad adecuada sin ser demasiado disruptiva para la búsqueda. En general, el algoritmo evolutivo propuesto demostró su capacidad para generar imágenes realistas y de alta calidad de rostros humanos, con el potencial de engañar a sistemas automáticos de reconocimiento facial.

### **Multiobjective evolutionary search of the latent space of generative adversarial networks for human face generation**

Correa y cols. (2023) se basaron en los trabajos previos de Machín y cols. (2021, 2022) para presentar una comparación de enfoques evolutivos para la generación de imágenes sintéticas de rostros humanos con múltiples objetivos basados en la exploración del espacio latente de las redes generativas antagónicas. El enfoque propuesto buscó generar imágenes de rostros humanos distintos pero con determinados atributos definidos. La optimización utilizó redes generativas antagónicas para la generación de rostros, reconocimiento facial para evaluar la similitud y los atributos de los individuos y un algoritmo evolutivo para explorar el espacio de búsqueda.

La propuesta se fundamentó en StyleGAN2 para generar imágenes de rostros humanos que cumplieran ciertos criterios, como la similitud con las imágenes objetivo, la raza, el género o la edad objetivo. La búsqueda de un espacio latente adecuado para generar imágenes que cumplan con estos criterios se realizó primero mediante un enfoque evolutivo mono-objetivo a partir de lo presentado por Machín y cols. (2021) y luego mediante un enfoque multi-objetivo basado en Machín y cols. (2022). El algoritmo evolutivo mono-objetivo evaluó la ponderación de la similitud entre las imágenes generadas y las imágenes objetivo y la

similitud del atributo de las imágenes generadas y las imágenes objetivo, mientras que el enfoque multi-objetivo utilizó la similitud entre el atributo elegido de los rostros generados y objetivos, y la similitud entre los rostros generados y los rostros objetivo. El modelo utilizado para evaluar la calidad de las imágenes fue FaceNet.

El enfoque evolutivo propuesto empleó una representación de solución que codificaba vectores en el espacio latente de StyleGAN2 como individuos compuestos por 512 números de punto flotante. La similitud se evaluó mediante la distancia coseno entre los embeddings de dos imágenes. La función de aptitud utilizó una combinación ponderada lineal de la distancia al rostro objetivo y la distancia al atributo objetivo.

Para iniciar la búsqueda en el espacio latente, se generó la población de soluciones mediante una distribución normal (gaussiana) con media cero y desviación estándar uno. La inicialización aleatoria proporcionó valores útiles para dar inicio a la búsqueda sin necesitar información precisa sobre las características del espacio latente. Al igual que Machín y cols. (2022) se optó por el operador de selección por torneo, con un tamaño de torneo de tres individuos, de los cuales solo uno sobrevivió como progenitor. Se utilizó también el operador de recombinación BLX- $\alpha$  (Blend- $\alpha$ ).

Se pudo comprobar que el algoritmo mono-objetivo limitaba la diversidad de los resultados, al eliminar individuos que no cumplían con una medida aunque en gran medida cumplieran con la otra medida. El algoritmo multi-objetivo logró equilibrar la elección de nuevos progenitores, colaborando con la diversidad en las soluciones.

### 3.4. Utilidad de los artículos relevados

Los artículos de generación de rostros relevados utilizaron la implementación de StyleGAN2 para la generación de imágenes. Resulta interesante el método de inicialización de las soluciones explicado en los trabajos de Machín y cols. (2022) y Correa y cols. (2023), con una distribución Gaussiana de manera de generar valores iniciales útiles sin conocer la relación entre los vectores y las características deseadas. Machín y cols. (2021) propusieron otro método de inicialización, partiendo de una distribución normal para generar una inicialización aleatoria. Ambos métodos se consideran para la inicialización de la población.

Además los artículos de Correa y cols. (2023) y Machín y cols. (2021, 2022) describieron el uso del operador de combinación BLX- $\alpha$ . Se considera una buena opción experimentar con el operador BLX- $\alpha$  y otros operadores a manera de comparar su efectividad para generar descendientes.

Finalmente, se destaca el uso en Galatolo y cols. (2021) del algoritmo evolutivo NSGA-II, que resulta una opción interesante al momento de definir el algoritmo y sus operadores en la solución propuesta en el proyecto.

La Tabla 3.1 muestra año, lugar, autores, nombre y concepto clave de los artículos analizados. Se dividen las entradas de la tabla en dos categorías correspondientes a las subsecciones 3.2 y 3.3.

<i>Año</i>	<i>Lugar</i>	<i>Autores</i>	<i>Nombre</i>	<i>Concepto Clave</i>
Uso de redes neuronales generativas antagónicas (GAN) y uso de algoritmos evolutivos para explorar el espacio latente de las GANs				
2022	Argentina	Pitiddu, Veiga	Diseño de clasificadores de dígitos manuscritos generados por redes GAN	Redes Neuronales Generativas Antagónicas
2020	Portugal	Fernandes, Correia, Machado	Evolutionary Latent Space Exploration of Generative Adversarial Networks	Prueba de concepto de la posibilidad de guiar la exploración del espacio latente de redes GAN a través de algoritmos evolutivos
2020	Francia	Roziere, Teytaud, Hosu, Lin, Rapin, Zameshina, Teytaud	EvoGAN: Evolutionary Generative Adversarial Networks	Podría ir como introducción al uso de algoritmos evolutivos para mejorar los vectores de la GAN
2021	Italia	Galatolo, Cimino, Vaglini	Generating Images from Caption and Vice Versa via CLIP-Guided Generative Latent Space Search	Generación de texto a partir de imágenes con algoritmo multi-objetivo y redes GAN, generación de imágenes a partir de texto con algoritmo mono-objetivo y redes GAN
Generación de rostros a través del uso de algoritmos evolutivos para la exploración del espacio latente de redes neuronales generativas antagónicas				
2018	Australia	Neumann, Pyromallis, Alexander	Evolution of Images with Diversity and Constraints Using a Generative Adversarial Network	Aplicación de optimización evolutiva para la mejora de la generación de imágenes con redes GAN
2021	Uruguay, España	Machín, Nesmachnow, Toutouh	Evolutionary latent space search for driving human portrait generation	Uso de algoritmos evolutivos mono-objetivo para la exploración del espacio latente de redes GAN en generación de rostros
2022	Uruguay, España	Machín, Nesmachnow, Toutouh	Multi-target evolutionary latent space search of a generative adversarial network for human face generation	Uso de algoritmos evolutivos multi-objetivo para la exploración del espacio latente de redes GAN en generación de rostros
2023	Uruguay, España	Correa, Mignaco, Rey, Machín, Nesmachnow, Toutouh	Multiobjective evolutionary search of the latent space of Generative Adversarial Networks for human face generation	Uso de algoritmos evolutivos mono-objetivo y multi-objetivo para la exploración del espacio latente de redes GAN en generación de rostros

Tabla 3.1: Descripción de estudios relacionados con GANs y algoritmos evolutivos

## Capítulo 4

# Descripción del problema e implementación de la solución

Este capítulo presenta la descripción del problema de la exploración del espacio latente para la generación de rostros humanos, el modelado y la implementación de la solución que se propone en el proyecto.

### 4.1. Descripción del problema

Las redes neuronales antagónicas trabajan sobre un espacio latente de vectores. El espacio latente es un espacio multidimensional donde cada vector representa una combinación de características latentes. Cuando los vectores se transforman a través de una red generativa, resultan en un ejemplo específico de los datos de entrada (por ejemplo, una imagen de un rostro humano). Se cumple que vectores cercanos en el espacio latente generan datos similares, por lo que resulta importante lograr una eficiente exploración del espacio para mantener la diversidad y a su vez reconocer posibles soluciones para buscar mejores opciones en su entorno. Garantizar que la exploración cubra adecuadamente el espacio latente para evitar regiones de baja calidad o sin sentido es crucial. Algunos métodos pueden quedarse atrapados en regiones locales, sin explorar otras áreas potencialmente valiosas. La exploración del espacio latente en muchos casos busca optimizar múltiples objetivos simultáneamente, y pequeñas variaciones en las coordenadas latentes pueden resultar en cambios significativos en las características generadas. La alta dimensionalidad del espacio a explorar (conformado por 512 dimensiones) y un dominio en los números reales, hace que una exploración eficiente del espacio se vuelva un problema de gran complejidad computacional.

Los algoritmos evolutivos multi-objetivo se presentan como una solución al problema de la exploración del espacio latente porque permiten la diversificación y optimización simultánea de múltiples objetivos. Al mantener una población de individuos permiten explorar diversas regiones del espacio latente a la vez, lo que ayuda a evitar que se queden atrapados en mínimos locales y garantiza una cobertura más completa del espacio. Además, pueden manejar varios objetivos, como la generación de imágenes realistas y la preservación de atributos específicos como raza, género y edad, encontrando un equilibrio adecuado entre ellos.

A lo largo del proyecto se diseñó y desarrolló un algoritmo evolutivo multi-objetivo que guía la generación del rostro humano mediante la exploración del espacio latente de codificaciones de atributos. El algoritmo busca optimizar tres objetivos simultáneamente: la similitud entre la imagen generada y la imagen objetivo, la alineación del rostro con un género elegido y la alineación del rostro con una raza elegida. El algoritmo evalúa múltiples individuos y evoluciona las poblaciones a lo largo de varias generaciones para converger en imágenes sintéticas que buscan cumplir con los tres criterios, manteniendo la diversidad y evitando soluciones sesgadas.

El uso de un enfoque multi-objetivo permite al algoritmo equilibrar múltiples prioridades, creando una gama de soluciones que reflejan diferentes grados de similitud y alineación de género y raza. A través de técnicas evolutivas, el algoritmo explora una amplia variedad de combinaciones de atributos, identificando las que mejor se adaptan a los criterios establecidos. El proceso de selección y mejora iterativa garantiza que las imágenes finales sean precisas y diversas.

El problema de generación sintética de rostros humanos es directamente aplicable a la verificación de la robustez de los sistemas automáticos de identificación facial. Proporciona medios para crear conjuntos de datos variados y balanceados sin comprometer la privacidad de los individuos. Además, permite probar los sistemas de identificación facial en una variedad de condiciones controladas y diversificadas, ayudando a desarrollar sistemas más precisos, justos y resistentes a variaciones en los datos de entrada.

## 4.2. Modelado de la solución

En esta sección se definen los componentes del algoritmo evolutivo: la representación de los individuos, la función de fitness, los operadores de selección y variación, y la inicialización de la población. Se describe cómo se han implementado los elementos necesarios para el diseño y desempeño del algoritmo.

### 4.2.1. Representación de los individuos

Los individuos son la entrada a la red neuronal generadora de imágenes, en este caso StyleGAN, por lo que se representan como vectores de 512 números reales, acorde al tamaño de entrada de StyleGAN. Cada vector o genotipo representa un punto en el espacio latente, y cada número representa un alelo del genotipo. Cada vector se corresponde con un único rostro o fenotipo en el espacio de posibles soluciones del problema.

### 4.2.2. Función de evaluación o fitness

Para definir la función de fitness se plantean tres objetivos: similitud de raza (SR), similitud de género (SG) y similitud de rostro o embedding (SE). La similitud de raza se calcula a partir de la distancia euclidiana entre el vector de razas del individuo y el vector objetivo. [Correa y cols. \(2023\)](#) propusieron que los valores devueltos por el modelo de análisis de DeepFace no resultan útiles al momento de calcular el fitness de los individuos, dado que devuelven valores binarios que reflejan si un individuo presenta una raza o no. De esta manera se hace imposible definir preferencias entre dos individuos que presentan la raza objetivo, aunque un individuo se encuentre más cercano al objetivo que otro. [Correa y cols. \(2023\)](#) plantearon la eliminación de la última capa de la red neuronal de DeepFace, utilizando en su lugar una función de normalización para nivelar los resultados de cada raza. El vector de razas del individuo se define a partir de la normalización de la predicción de razas proporcionada por el modelo de razas de DeepFace, que equivale al uso del modelo de análisis de DeepFace sin la última capa. Dada la predicción de razas  $P_r$ , el vector de razas  $V_r$  se calcula a través de las ecuaciones 4.1 y 4.2. Dado el vector de razas objetivo  $V_o$ , la similitud de razas  $SR$  se calcula como  $SR = euclidian\_distance(V_r, V_o)$ .

$$P'_r = \frac{P_r - \min(P_r)}{\max(P_r) - \min(P_r)} \quad (4.1)$$

$$V_r = \frac{P'_r}{\sum(P'_r)} \quad (4.2)$$

Considerando los cambios realizados por [Correa y cols. \(2023\)](#) sobre el vector de razas, se realiza un estudio similar respecto al vector de géneros que se desarrolla en la [Subsección 4.3.3](#). A partir del estudio se decide definir el vector de géneros del individuo a partir de aplicar una capa de función softmax a la predicción de géneros proporcionada por el modelo de géneros de DeepFace. Entonces dada la predicción de géneros  $P_g$  y el vector objetivo  $V_o$ , la similitud de género  $SG$  se calcula como  $SG = cosine\_distance(V_g, V_o)$ , donde  $V_g$  corresponde al vector de predicción de géneros calculada a través de la [Ecuación 4.3](#).

$$V_g = \frac{1}{1 + e^{-0,3 \cdot P_g}} \quad (4.3)$$

Por último, la similitud de rostro se calcula como la distancia coseno entre los embeddings del individuo ( $E_i$ ) y los embeddings del rostro objetivo ( $E_o$ )  $SE = \text{cosine\_distance}(E_i, E_o)$ .

El algoritmo busca minimizar simultáneamente todas estas métricas de similitud, ( $\min(SR), \min(SG), \min(SE)$ ), para obtener individuos que correspondan con las características de raza, género y rasgos faciales del objetivo.

### 4.2.3. Selección

En esta sección se detallan los métodos de selección utilizados en el algoritmo planteado. La selección de padres se realiza con selección por torneo basado en dominancia, mientras que para la selección de supervivientes se utiliza la selección utilizada en NSGA-II (Deb y cols., 2002).

#### Selección por torneo

La selección por torneo consiste en completar el conjunto de padres a través de competencias de dos entre los posibles individuos. Para esto se seleccionan dos individuos al azar y un número cualquiera  $r \in [0, 1]$ . Dado un parámetro  $k$ , si  $r < k$  entonces el individuo con mejor fitness se selecciona para ser uno de los padres, y si  $r \geq k$  se selecciona el individuo con peor fitness para ser padre. Luego ambos individuos se devuelven al conjunto original donde pueden volver a ser elegidos para competir. El proceso se repite hasta completar el conjunto de padres (Mitchell, 1996). La selección por torneo basado en dominancia define cual es el mejor individuo entre los dos seleccionados para competir de acuerdo a la dominancia de uno sobre el otro. Si ambos se dominan respectivamente se utiliza la distancia de crowding para definir el mejor individuo. El valor del parámetro  $k$  utilizado en este proyecto fue 0.5. El algoritmo garantiza que cada individuo será elegido como padre un máximo de dos veces asumiendo que los individuos originales son únicos y que empezando desde el comienzo todos los padres consecutivos serán diferentes (Developers, 2024).

La selección por torneo permite que los individuos más aptos sean seleccionados como padres y prosperen a través de las generaciones, a la vez que facilita la diversidad de los individuos seleccionando en algunos casos el individuo menos apto como padre de la nueva generación.

#### Selección de NSGA-II

NSGA-II es un algoritmo genético multi-objetivo presentado en 2002 que define, entre otros, un método de selección que crea un conjunto de padres combinando las poblaciones de padres e hijos y seleccionando las mejores  $N$  soluciones (con respecto al fitness y la diversidad). Se utilizan para la selección de supervivientes los frentes de Pareto y la distancia de crowding, agregando los individuos de cada frente ordenados por distancia de crowding al grupo de supervivientes hasta que se alcance la cantidad deseada.

#### 4.2.4. Operadores de variación

Siguiendo el algoritmo diseñado por [Correa y cols. \(2023\)](#), se utiliza el método de blend-alpha como operador de cruce, con  $\alpha = 0,5$  y el método de mutación gaussiana como operador de mutación, con  $\mu = 0$  y  $\sigma = 1$ .

##### Blend- $\alpha$

El operador de cruce de Blend- $\alpha$ , presentado por [Eshelman y Schaffer \(1993\)](#), es un método de recombinación que utiliza un parámetro  $\alpha$  para definir la dimensión de la distancia entre los hijos generados y sus padres. A diferencia de los operadores de cruce simples, que utilizan los valores de los alelos de uno u otro de los padres para generar los alelos de los hijos, el método de blend- $\alpha$  toma el valor del alelo de cada uno de sus padres para generar un intervalo. El intervalo generado depende de  $\alpha$  y se utiliza para generar el alelo correspondiente en el hijo. Dado un valor  $\alpha \in [0, 1]$ , y dos padres  $A = (a_1, a_2, \dots, a_n)$  y  $B = (b_1, b_2, \dots, b_n)$ , cada alelo  $h_i$  del hijo H es un número aleatorio elegido uniformemente del intervalo  $[\text{mín}(a_i, b_i) - \alpha * |a_i - b_i|, \text{máx}(a_i, b_i) + \alpha * |a_i - b_i|]$ .

El método de Blend- $\alpha$  permite crear hijos diferentes entre sí y diferentes a los padres, encontrando vectores intermedios entre los padres. Permite una mayor exploración del espacio latente al introducir mayor variabilidad en cada uno de los alelos.

##### Mutación gaussiana

El método de mutación gaussiana es uno de los más utilizados en algoritmos genéticos para introducir variabilidad a la población de individuos. Se basa en el empleo de la probabilidad gaussiana (o probabilidad normal) para modificar algunos de los alelos de un individuo. Para mutar un individuo primero se decide para cada alelo, con una probabilidad definida, si corresponde aplicar la mutación. En caso de aplicar la mutación en un alelo, se modifica sumando un valor elegido con probabilidad gaussiana de media  $\mu = 0$  y desviación estándar  $\sigma$  al valor original del alelo. El valor  $\sigma$  determina la distancia entre los alelos modificados y los originales, un mayor valor de  $\sigma$  provoca mayores cambios en los valores de los alelos de un individuo. La mutación gaussiana permite generar nuevos individuos a partir de un único padre al generar cambios aleatorios en los alelos, colaborando con la diversidad en la población y evitando el estancamiento en óptimos locales.

#### 4.2.5. Población

La población inicial se construye a partir de individuos seleccionados de manera aleatoria siguiendo una distribución normal  $\mathcal{N}(0, 1)$ . La población evoluciona por un total de 100 generaciones, sometándose a selecciones, cruzamientos y mutaciones.

## 4.3. Arquitectura del pipeline

En esta sección se presenta la arquitectura del pipeline utilizado en el proyecto. Se describe la estructura y funcionamiento general del pipeline, su organización y la manera en que opera. Se detallan los principales componentes que lo integran, incluyendo DEAP, StyleGAN, RetinaFace, Facebook DeepFace, y el Analizador.

### 4.3.1. Estructura y funcionamiento general del pipeline

En esta subsección se describe la estructura y funcionamiento general del pipeline.

#### Estructura del pipeline

El pipeline del proyecto está constituido por tres componentes principales.

- StyleGAN3: Componente utilizado para la generación de rostros a partir de los individuos con los que se trabaja.
- DeepFace: Framework utilizado para la alineación, reconocimiento y análisis de los rostros, proceso empleado para el cálculo del fitness de cada individuo.
- DEAP: Framework se utiliza para implementar el algoritmo evolutivo, actuando como un wrapper para los demás componentes.

Asimismo, se implementaron funciones auxiliares para el preprocesamiento de imágenes, cálculos de distancia y transformación de outputs. El pipeline fue implementado en su totalidad en Python, ya que los principales frameworks utilizados se encontraban disponibles en dicho lenguaje.

#### Funcionamiento del pipeline

Todo individuo perteneciente a la población de una generación en particular, se somete al siguiente proceso:

1. Su imagen correspondiente es generada por StyleGAN3, devolviéndola en representación matricial.
2. La imagen generada es preprocesada para reducir su tamaño.
3. A continuación, la imagen es analizada por el reconocedor, RetinaFace, que se encarga de centrar y recortar el rostro. Si el reconocedor es capaz de reconocer un rostro en la imagen, continúa con el resto del proceso; de lo contrario, se aplica una penalización al valor de fitness con el fin de que el individuo no sea considerado para futuras generaciones.

4. Dado un rostro válido, se realizan tres acciones:
  - 4.1. Se utiliza la red de Facebook DeepFace para analizar la imagen y generar su embedding.
  - 4.2. Se utiliza el framework DeepFace para generar el vector de raza de la imagen.
  - 4.3. Se utiliza el framework DeepFace para generar el vector de género de la imagen.
5. Se calculan las distancias entre los tres vectores generados y los vectores objetivo, devolviendo estas distancias como el fitness final del individuo.
6. Una vez que se tienen valores de fitness para cada individuo, se realiza un proceso de selección de los mejores individuos, los cuales posteriormente podrán ser cruzados y/o mutados para formar una nueva generación.
7. Para cada individuo, se repite el ciclo desde el paso 1.

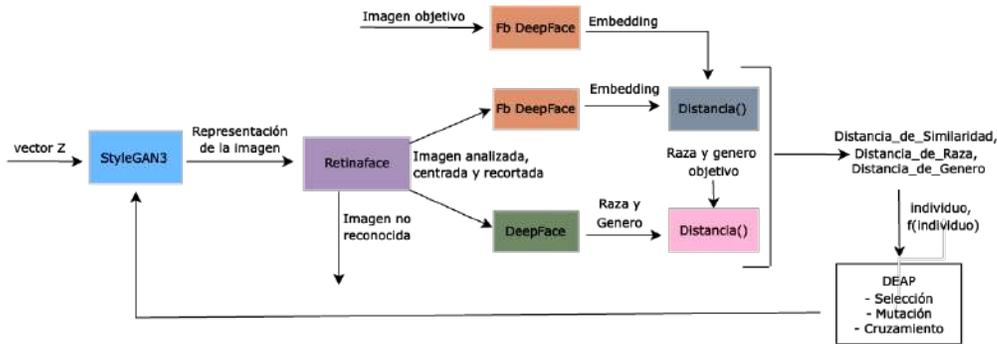


Figura 4.1: Funcionamiento del Pipeline

#### 4.3.2. Funcionamiento de los principales componentes del pipeline

En esta sección se presenta el funcionamiento de los principales componentes del pipeline. Los componentes presentados son DEAP, StyleGAN, RetinaFace, Facebook DeepFace y el analizador.

## DEAP

La implementación del algoritmo evolutivo se realizó utilizando las diferentes herramientas provistas por el framework DEAP. La estrategia evolutiva se compone de seis etapas principales, las cuales se detallan en el [Algoritmo 1](#) y la [Figura 4.2](#).

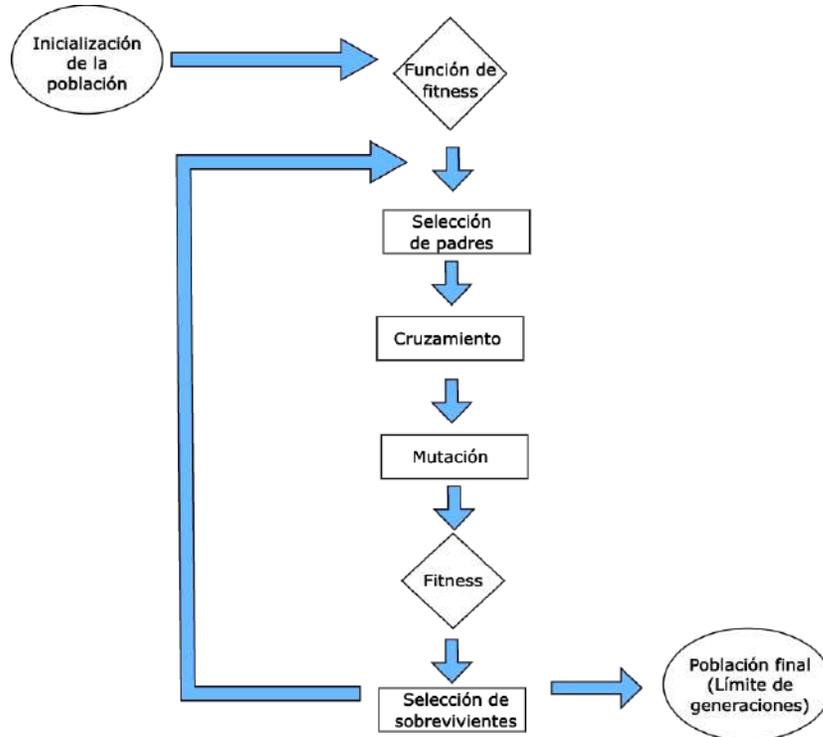


Figura 4.2: Etapas del Pipeline

Para la selección de la población inicial los individuos se inicializan de manera aleatoria, donde a cada uno de sus genes se le asigna un valor inicial según una distribución normal  $\mathcal{N}(\mu, \sigma)$ . Dada una población de individuos de tamaño  $X$ , se seleccionan los padres de la siguiente generación utilizando un selector por torneo basado en dominancia. Los  $X$  individuos seleccionados, entre los cuales un mismo individuo puede repetirse varias veces, conformaron el conjunto de padres de los cuales se toman parejas para su cruzamiento.

Para el cruzamiento de los individuos, se utiliza el operador BLX- $\alpha$ , con el valor  $\alpha$  seteado en 0.5. El operador se distingue por no solo combinar a los padres sino también explorar el espacio cercano a ellos.

La mutación se realiza con el operador de mutación gaussiana con media 0 y desviación estándar 1. Los nuevos individuos se evalúan según una función de fitness multi-objetivo, cuya implementación se expone en la [Subsección 4.2.2](#).

Una vez generados los descendientes y obtenido su valor de fitness, se utiliza el algoritmo de selección de NSGA-II para generar una nueva población a partir de la unión de la generación padre y sus descendientes. El ciclo se repite hasta cumplir la cantidad establecida de generaciones.

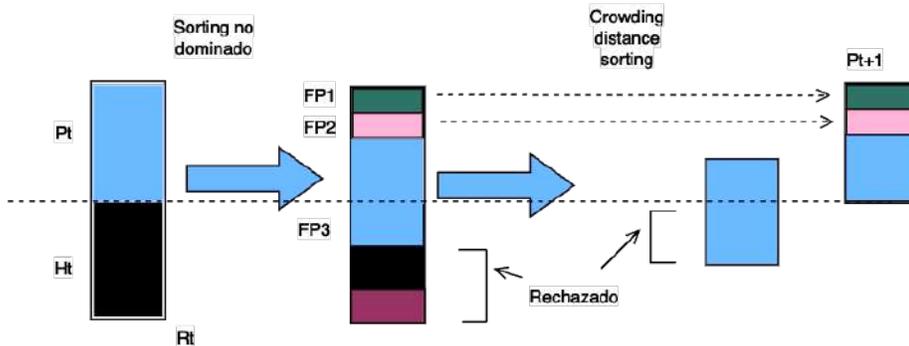


Figura 4.3: Selección de NSGA-II

## StyleGAN

El componente de StyleGAN3 se encarga de la generación de imágenes. Toma un vector de 512 números reales, donde cada vector determina de manera exclusiva una imagen, por lo que el modelo se considera determinista. Las imágenes generadas son representadas por una tupla con el canal de 3 colores (RGB), la altura y el ancho de la imagen representada, por lo que tiene dimensiones (3,1024,1024).

En una primera implementación, las imágenes generadas por la red eran guardadas en disco para después ser leídas por DeepFace. Sin embargo, con el fin de acelerar el funcionamiento del pipeline, se optó por manejar las imágenes en su representación matricial, evitando el overhead generado por escritura y lectura en disco. Al realizar el cambio, fue necesario considerar que DeepFace trabaja con un canal BGR en lugar de RGB. Por lo tanto, después de generar las imágenes y obtener sus representaciones, se intercambiaron los vectores correspondientes al primer y tercer canal de la tupla para asegurar que el reconocedor y analizador puedan procesar correctamente las imágenes en el formato BGR requerido.

El modelo del generador StyleGAN3 utilizado es el modelo pre entrenado stylegan3-r-ffhq-1024x1024. El modelo fue entrenado en base al dataset de imágenes Flickr-Faces-HQ (ffhq) (Karras y cols., 2019), con resoluciones de  $1024 \times 1024$  píxeles, pudiendo generar así imágenes de la misma resolución. Es equivalente a su vez en cuanto a traslación y rotación, a diferencia de su versión anterior StyleGAN2. StyleGAN2 mantenía ciertos píxeles constantes en imágenes al explorar sus vecinos en el espacio latente. Por otro lado StyleGAN3 logra que las características representadas puedan trasladarse y rotar en conjunto con

las imágenes generadas. El hecho de que Stylegan3 sea invariante ante la traslación y rotación permite que los atributos representados vayan a la par del movimiento del rostro que se busca representar, evitando el efecto de aliasing en las imágenes generadas que produce StyleGAN2.

La invarianza ante traslación y rotación de Stylegan3 es beneficiosa a la hora de explorar el espacio latente utilizando algoritmos evolutivos, ya que las mutaciones y cruzamientos realizados sobre la población generan muchas veces variaciones en el posicionamiento de las imágenes. Si los píxeles se mantuvieran estáticos como ocurre con StyleGAN2, diferentes atributos de los sujetos representados se mantendrían en una posición antinatural a medida que el rostro rota, generando imágenes de peor calidad.

### RetinaFace

RetinaFace es el componente encargado de realizar la detección de rostros. Para ello, las imágenes son en primer lugar convertidas a una resolución de  $224 \times 224$  píxeles. Una vez realizada la reducción, RetinaFace se encarga de identificar las coordenadas del rostro que se encuentra en la imagen, así como algunos de sus puntos de referencia principales.

Si un rostro es reconocido dentro de la imagen, el componente realiza un recorte, alineando el rostro en el centro de la imagen que posteriormente es analizada por los componentes restantes. Además, se realiza una expansión sobre el rostro con un porcentaje del 40%, debido a que los análisis de la [Subsección 5.1.5](#) revelaron que la expansión aumentaba la fidelidad en la correcta detección de la raza y el género. Por otro lado, si un rostro no es detectado dentro de la imagen, se le aplica a dicho individuo una penalización en su valor de fitness.

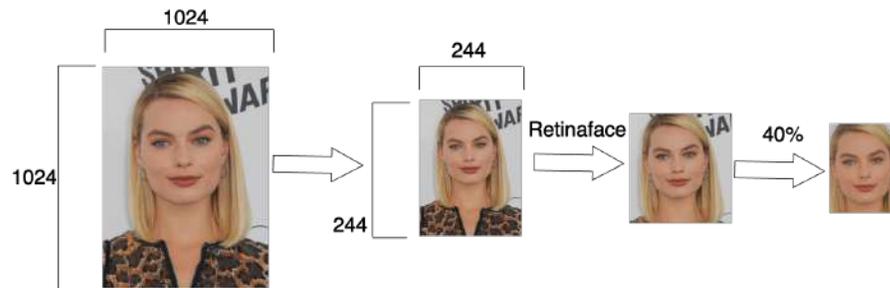


Figura 4.4: Funcionamiento de RetinaFace

### Facebook DeepFace

Facebook DeepFace es el analizador encargado de extraer los embeddings del rostro representado por un individuo. La representación se realiza con el fin de poder evaluar la similitud con el rostro objetivo, cuyo embedding es extraído por única vez al comenzar el algoritmo.

Los embeddings son vectores reales de 4096 dimensiones. Para identificar la similitud de un rostro con otro, los dos vectores que los representan son comparados según una métrica de distancia. Cuanto más cercanos en el espacio están estos vectores, mayor es la probabilidad de que los rostros comparados pertenezcan a la misma persona. Cada uno de los embeddings de los individuos se compara con el de la imagen objetivo utilizando la distancia coseno.

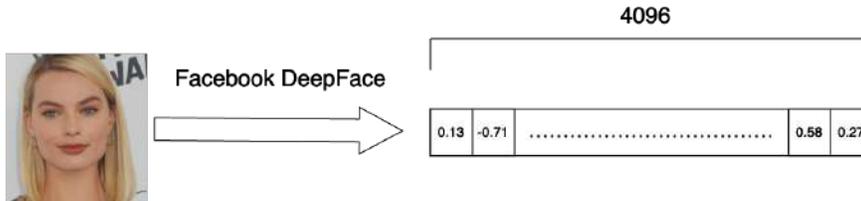


Figura 4.5: Transformacion de una imagen en embeddings

### Analizador

Para la predicción de raza y de género, se utilizan algoritmos con modelos pre entrenados del framework DeepFace. Como modelo base de ambos analizadores se utiliza VGG-Face, en el cual se fijan los pesos en las primeras capas debido a su capacidad de detección de atributos faciales. En el predictor de razas, todas las capas menos las últimas siete son fijadas, mientras que en el de género se fijan todas menos las últimas tres.

Ambos módulos toman como entrada una imagen representada por el vector (224,224,3) por lo que el re-dimensionamiento realizado por RetinaFace en etapas anteriores es también útil en este paso. Además, los valores de representación de los píxeles son normalizados, dividiendo el valor de RGB de cada píxel entre 255, para generar a un valor entre 0 y 1.

El analizador para razas de DeepFace fue entrenado a partir de la base de datos etiquetada FairFace (Karkkainen y cols., 2021). El analizador cuenta con seis salidas, una por cada raza, que fueron modificadas como se detalla en la Subsección 4.2.2. La capa de salida del analizador de géneros consta tan solo de dos clases, cada una correspondiente a un género, que también fueron modificadas como se detalla en la Subsección 4.3.3. Para el entrenamiento del analizador de género se utilizó el dataset IMDB-WIKI (Rothe y cols., 2016).

Por último, para calcular la distancia entre el género objetivo y el de un individuo se utiliza la distancia coseno y para la raza se utiliza la distancia euclidiana.

### 4.3.3. Eliminación de la capa softmax de DeepFace para el género

El algoritmo evolutivo busca que se optimicen tres métricas simultáneamente. La primera métrica evalúa la similitud entre la cara generada y la imagen objetivo. La segunda y la tercera métrica evalúan la diferencia en los atributos estudiados (género y raza) entre la imagen generada y la imagen objetivo.

Cuando DeepFace evalúa las imágenes, genera un vector de probabilidad  $X_g = [x_0, x_1]$  para los géneros después de aplicar la última función de activación softmax, antes de la capa de salida. Sin embargo, el vector de probabilidad resultado de la activación softmax no resulta útil como función objetivo debido a un desequilibrio significativo en los valores. Los valores tienden a estar extremadamente cerca de 1 para el género predicho y cerca de 0 para el otro género, causando que los vectores de fitness generados sean muy similares entre individuos. Si todos los valores  $X_g$  son muy cercanos a  $[0, 1]$  o  $[1, 0]$ , buscar nuevos individuos cuyo fitness minimice el valor  $X_g - \{[1, 0], [0, 1]\}$  es cada vez más complejo.

El proyecto propuso aplicar un enfoque diferente, experimentando con dos opciones. Como primera opción se eliminó la función de activación softmax final aplicada por DeepFace. Los valores en el vector se normalizaron, aplicando la [Ecuación 4.4](#) para obtener un arreglo de valores en el intervalo  $[0, 1]$  que cumpla que la suma de todas sus entradas sea 1.

$$\widehat{X}_g = \frac{x_i + \sum_j X_{g_j}}{\sum_j X_{g_j}} \quad (4.4)$$

La segunda opción presentada propuso sustituir la función softmax por dos funciones sigmoideas para hallar la probabilidad de cada género de manera independiente, y luego se normalizaron los valores aplicando la [Ecuación 4.5](#).

$$\widehat{X}_g = \left[ \frac{x_f}{x_f + x_m}, \frac{x_m}{x_f + x_m} \right] \quad (4.5)$$

Con el fin de determinar la distribución de la clasificación realizada por DeepFace al utilizar softmax, se creó una base de 300 imágenes, conteniendo mujeres, hombres y personas andróginas en igual proporción. A continuación, se analizaron los rostros seleccionados con el fin de obtener la clasificación de sus géneros según DeepFace, utilizando las tres medidas propuestas.

La [Figura 4.6](#) muestra la distribución de los valores antes de quitar la función de activación softmax, transformando los valores de probabilidad en porcentaje. La [Figura 4.7](#) muestra acercamientos de la [Figura 4.6](#) en los intervalos  $[0, 10]$  y  $[90, 100]$ . La mayoría de los valores se encuentran en los extremos del intervalo, mayoritariamente en los subintervalos  $[0, 1]$  y  $[99, 100]$ . Al utilizar softmax, DeepFace clasificó el género de los rostros mayoritariamente de manera binaria, sin tomar en cuenta rostros que se encontraban en puntos medios en cuanto al género.

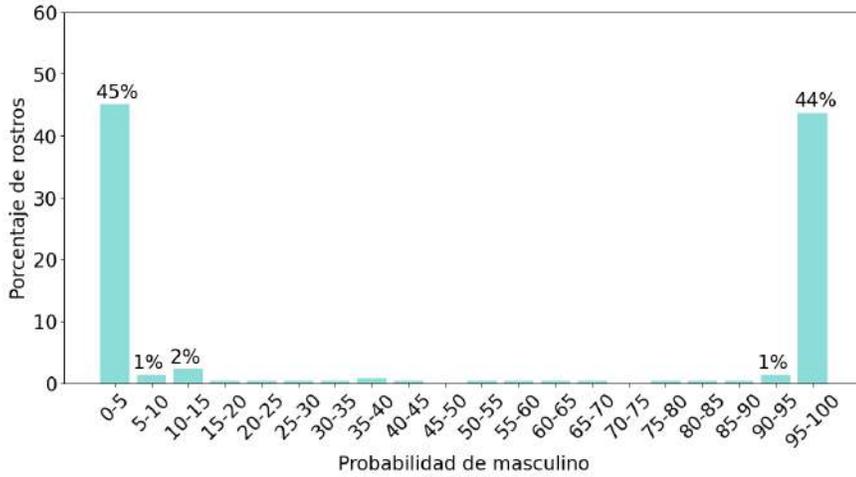
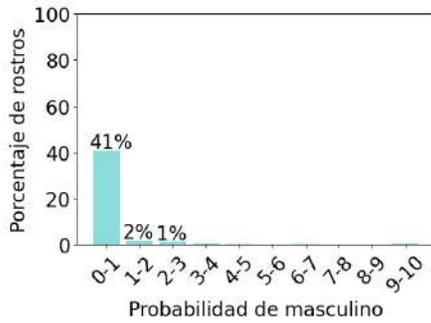
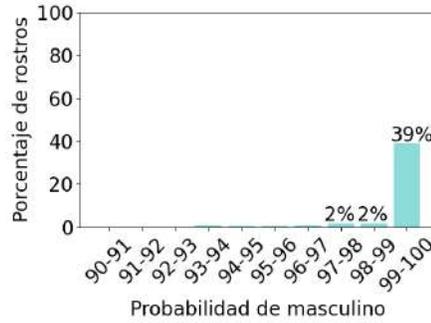


Figura 4.6: Distribución de los valores antes de quitar la función de activación softmax



(a) Corte 1: Distribución de los valores antes de quitar la función de activación softmax



(b) Corte 2: Distribución de los valores antes de quitar la función de activación softmax

Figura 4.7: Distribución de los valores antes de quitar la función de activación softmax

La [Figura 4.8](#) detalla los resultados luego de realizar la normalización descrita en la [Ecuación 4.4](#). Los valores ya no se encuentran en los extremos de la gráfica, sin embargo se acumulan en los subintervalos  $[15, 25]$  y  $[70, 75]$ . Los valores no están distribuidos de manera consistente con los rostros seleccionados, sino que se concentran en algunas regiones. La distribución no consistente de los valores también se presenta en la utilización de softmax.

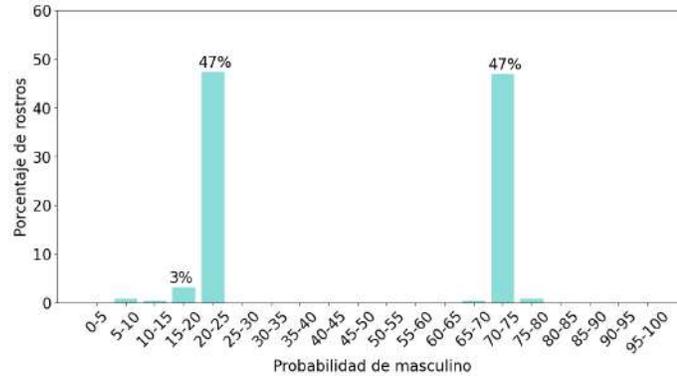


Figura 4.8: Distribución de los valores después de aplicar la normalización

Las figuras [4.9](#) y [4.10](#) muestran las distribuciones generadas al aplicar la función sigmoide detallada en la [Ecuación 4.6](#). Se evidencia que la sustitución de la función de activación softmax por una función sigmoide permite una distribución consistente con los rostros utilizados para el análisis, en contraste con la concentración previa en los extremos o en determinados intervalos.

$$x = \frac{1}{1 + e^{(-\alpha \cdot x)}} \quad (4.6)$$

Después de evaluar diferentes valores para el parámetro  $\alpha$  de la función sigmoide, se decidió utilizar  $\alpha = 0,3$ . Se observó que con el valor seleccionado la distribución en la gráfica fue más congruente con los datos de entrada, en comparación con el resto de los valores. En las figuras [4.9](#) y [4.10](#), se observa que con  $\alpha = 0,5$  los valores se mantienen más cerca de los extremos, mientras que con  $\alpha = 0,3$  los valores se encuentran distribuidos de forma consistente con la distribución del dataset utilizado.

En resumen, el enfoque propuesto demostró ser efectivo en mejorar la distribución de los valores de probabilidad de género generados por DeepFace. La eliminación de la función de activación softmax y la aplicación de una función sigmoide en su lugar permitió obtener una distribución más uniforme de los valores.

Con el fin de validar la nueva distribución de géneros generada a través de la aplicación de la función sigmoid, se buscó determinar si existía una correlación entre la asignación de género de la función sigmoid y el género real de las imágenes. Se definieron cinco rangos basados en la probabilidad de que el rostro

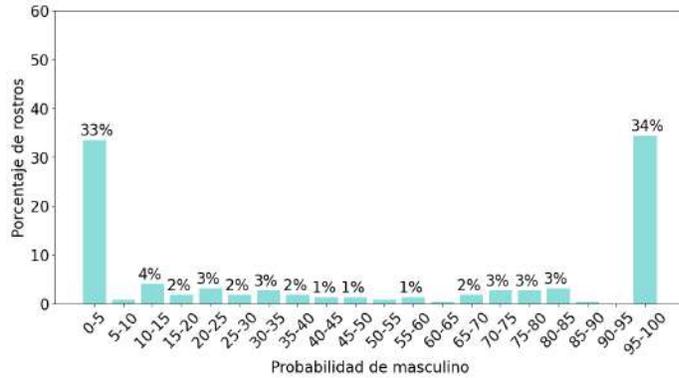


Figura 4.9: Distribución de los valores aplicando la función de sigmoid con  $\alpha = 0,5$

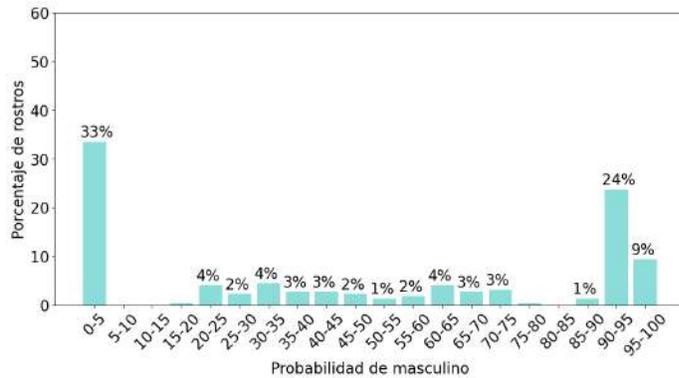


Figura 4.10: Distribución de los valores aplicando la función de sigmoid con  $\alpha = 0,3$

pertenezca al género masculino asignada por la función sigmoid. Los rangos analizados fueron  $R_m$  (rostros clasificados con probabilidad entre 99% y 100% de ser hombres),  $R_{m75}$  (rostros clasificados con probabilidad entre 74.9% y 75.1% de ser hombres),  $R_a$  (rostros clasificados con probabilidad entre 49.9% y 50.1% de ser hombres),  $R_{m25}$  (rostros clasificados con probabilidad entre 24.9% y 25.1% de ser hombres) y  $R_f$  (rostros clasificados con probabilidad entre 0% y 1% de ser hombres).

Se tomaron 150 imágenes aleatorias para cada uno de los rangos con el fin de analizar y etiquetar el género manualmente. La [Figura 4.11](#), [Figura 4.12](#) y [Figura 4.13](#) presentan algunos de los rostros analizados según cada rango seleccionado. Para cada rango establecido, se determinó si cada imagen correspondía a alguna de las categorías presentadas en la [Tabla 4.1](#).



(a) Rostros clasificados con un 99 %-100 % de certeza de ser mujeres.



(b) Rostros clasificados con un 74.90 %-75.10 % de certeza de ser mujeres.

Figura 4.11: Ejemplos de rostros clasificados como mujeres.



(a) Rostros clasificados con un 99%-100% de certeza de ser hombres.



(b) Rostros clasificados con un 74.90%-75.10% de certeza de ser hombres.

Figura 4.12: Ejemplos de rostros clasificados como hombres.



(a) Rostros clasificados con un 49.90%-50.10% de certeza de ser hombres.

Figura 4.13: Ejemplo de rostro clasificado como andrógino.

<i>Nombre</i>	<i>Descripción</i>
Masculino	El rostro en la imagen es claramente identificable como masculino
Niño	El rostro en la imagen es claramente identificable como un niño varón
Andrógino	El rostro no es claramente identificable como masculino o femenino
Andrógino niño	El rostro no es identificable como masculino o femenino debido a que la persona presentada es un infante
Niña	El rostro en la imagen es claramente identificable como una niña mujer
Femenino	El rostro en la imagen es claramente identificable como femenino

Tabla 4.1: Descripción de categorías de identificación de rostros

La [Tabla 4.2](#) presenta el número de imágenes que fueron clasificadas en cada categoría (presentada en [Tabla 4.1](#)) para cada rango establecido. La [Tabla 4.3](#) y la [Figura 4.14](#) muestran la distribución porcentual de cada categoría según los rangos seleccionados. La distribución generada se condice con los valores asignados por DeepFace al utilizar la nueva función sigmoide. Los rostros clasificados como andróginos, en general presentaban rostros difíciles de reconocer como femenino o masculino. Si los rostros eran clasificados como masculinos, correspondían generalmente a adultos del género masculino, y los rostros clasificados como femeninos correspondían mayoritariamente a adultos de género femenino. El rango  $R_{m75}$  presentó en pocas ocasiones rostros clasificados como femeninos, y el rango  $R_{m25}$  presentó muy pocos rostros clasificados como de género masculino. Ambos rangos presentaron una mayor cantidad de rostros clasificados como andróginos que los rangos  $R_m$  y  $R_f$ .

<i>Clasificación</i>	$R_m$	$R_{m75}$	$R_a$	$R_{m25}$	$R_f$
Masculino	127	98	11	0	0
Niño	23	17	10	0	0
Andrógino	0	29	46	35	2
Andrógino niño	0	4	29	12	0
Niña	0	0	9	26	14
Femenino	0	2	45	77	134

Tabla 4.2: Distribución de clasificaciones según categorías de rostro

Por último, se buscó comparar las distribuciones generadas por las funciones de softmax, la [Ecuación 4.4](#) y la [Ecuación 4.5](#) para un conjunto aleatorio de imágenes generadas por StyleGAN3. La [Figura 4.15](#) muestra la distribución de

<i>Clasificación</i>	$R_m$	$R_{m75}$	$R_a$	$R_{m25}$	$R_f$
Masculino	84,7 %	65,4 %	7,3 %	0,0 %	0,0 %
Niño	15,3 %	11,3 %	6,7 %	0,0 %	0,0 %
Andrógino	0,0 %	19,3 %	30,8 %	23,3 %	1,3 %
Andrógino niño	0,0 %	2,7 %	19,3 %	8,0 %	0,0 %
Niña	0,0 %	0,0 %	6,0 %	17,3 %	9,3 %
Femenino	0 %	1,3 %	30,0 %	51,4 %	89,4 %

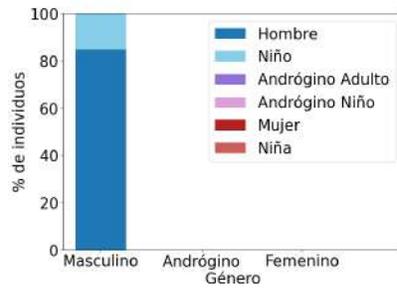
Tabla 4.3: Distribución porcentual de clasificaciones según categorías de rostro

los valores antes de quitar la función de activación softmax, transformando los valores de probabilidad en porcentaje. La mayoría de los valores se encuentran en los extremos del intervalo, mayoritariamente en los subintervalos  $[0, 1]$  y  $[99, 100]$ , al igual que en la [Figura 4.6](#).

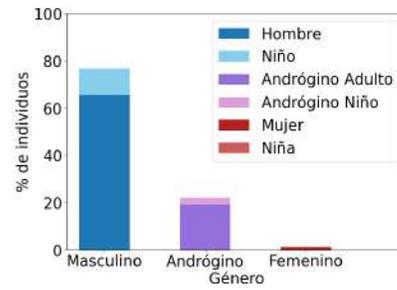
La [Figura 4.16](#) detalla los resultados luego de realizar la normalización descrita en la [Ecuación 4.4](#). Los valores se concentran en los subintervalos  $[20, 25]$  y  $[70, 75]$  como en la [Figura 4.8](#).

Las figuras [4.17](#) y [4.18](#) muestran las distribuciones generadas al utilizar la función sigmoide, detallada en la [Ecuación 4.6](#). Se evidencia una distribución más uniforme de los valores del vector de probabilidad a lo largo de todo el intervalo, logrando una dispersión más equitativa de los valores.

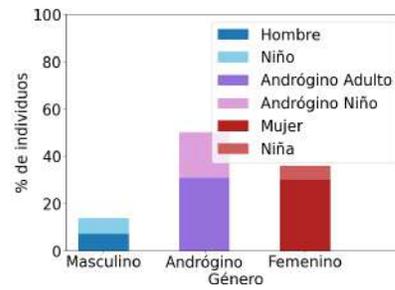
Se observó que con el valor  $\alpha$  seleccionado, la distribución en la gráfica fue más dispersa y uniforme a lo largo de todo el intervalo  $[0, 100]$ , en comparación con el resto de los valores. En las figuras [4.17](#) y [4.18](#), se observa que con  $\alpha = 0,5$  algunos valores se mantienen cerca de los extremos, mientras que con  $\alpha = 0,3$  los valores se encuentran mejor distribuidos.



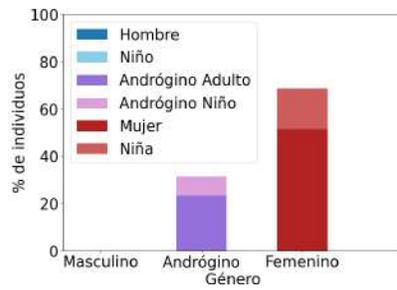
(a) Distribución porcentual de la categoría  $R_m$



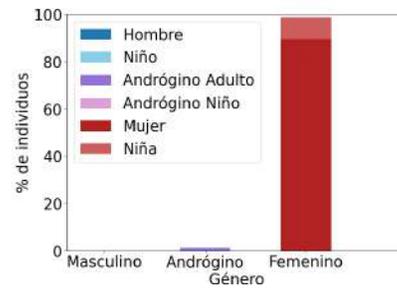
(b) Distribución porcentual de la categoría  $R_{m75}$



(c) Distribución porcentual de la categoría  $R_a$



(d) Distribución porcentual de la categoría  $R_{m25}$



(e) Distribución porcentual de la categoría  $R_f$

Figura 4.14: Distribución porcentual de clasificaciones según categorías de rostro

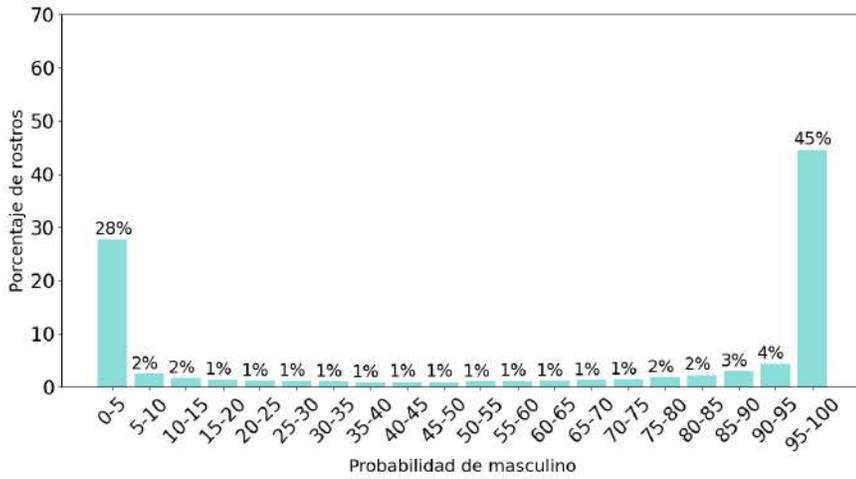


Figura 4.15: Distribución de los valores antes de quitar la función de activación softmax

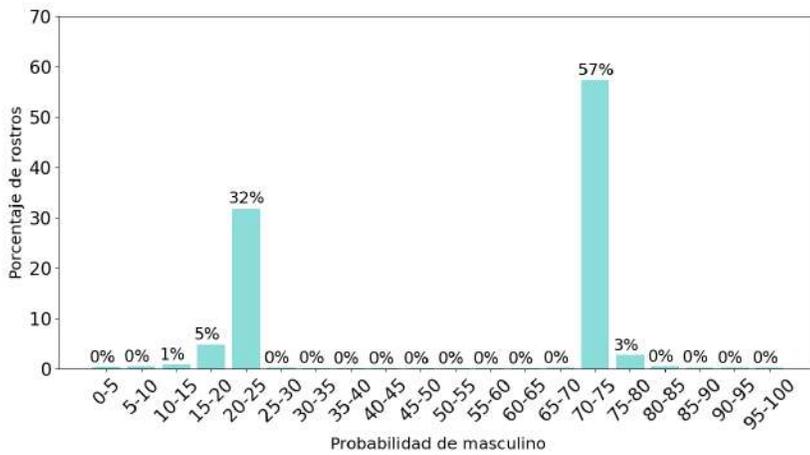


Figura 4.16: Distribución de los valores después de aplicar la normalización

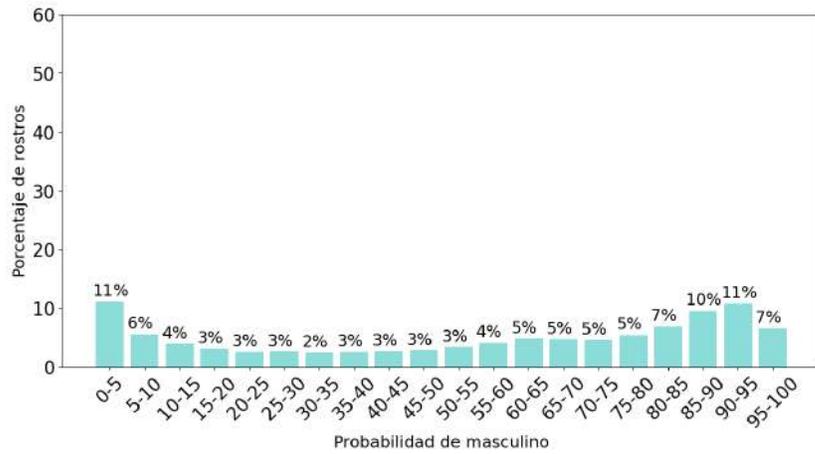


Figura 4.17: Distribución de los valores aplicando la funcion de sigmoid con  $\alpha = 0,5$

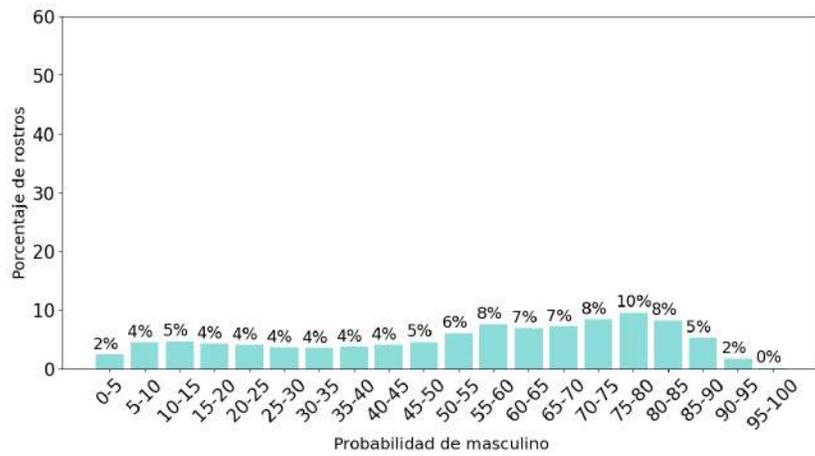


Figura 4.18: Distribución de los valores aplicando la funcion de sigmoid con  $\alpha = 0,3$

# Capítulo 5

## Experimentación

En este capítulo se presenta la metodología para el análisis experimental y los resultados del análisis experimental. Abarca la selección del modelo de detección, el análisis del sesgo en la generación de rostros y la elección de rostros para la experimentación. Además, se detallan los procedimientos de elección de la configuración paramétrica y la validación.

### 5.1. Metodología para el análisis experimental

En esta sección se presentan detalladamente las metodologías seguidas a lo largo del análisis experimental. Se presentan las metodologías para la selección del modelo de detección, el análisis de la exploración del espacio de vectores de entrada de StyleGAN3, el análisis de sesgo en la generación y evaluación de rostros, la metodología para la elección de la configuración paramétrica y la metodología para la validación.

#### 5.1.1. Metodología para la selección del modelo de detección y el modelo de análisis

Se generaron 700.000 imágenes con StyleGAN3 utilizando vectores latentes aleatorios. Para cada imagen generada, se utilizó un detector para detectar y recortar el rostro dentro de la imagen. En un principio se utilizó el modelo OpenCV para la tarea de detección, debido a que es el modelo predeterminado de DeepFace. Como se detalla en la [Subsección 5.2.1](#) OpenCV demostró una gran tasa de errores en la detección de rostros, por lo que otros modelos de detección fueron analizados para reemplazarlo. Con el fin de determinar un nuevo modelo de detección para el proyecto se realizó un estudio sobre SSD, MTCNN, RetinaFace, MediaPipe y Yunet. Se buscó un aumento de la capacidad de detección de rostros que no conlleve un aumento significativo en los tiempos de ejecución.

Para seleccionar el modelo de análisis facial a utilizar, se realizó una comparación sobre tiempos de ejecución de los principales analizadores provistos por la biblioteca DeepFace.

### 5.1.2. Metodología para el análisis de la exploración del espacio de vectores de entrada de StyleGAN3

Con el fin de verificar que la muestra de imágenes generadas en la [Subsección 5.1.1](#) se encuentran distribuidas en el espacio de posibles entradas de StyleGAN, se buscó representar las posiciones de los vectores de entrada que se usaron para generar las imágenes en el espacio de manera gráfica. La distribución uniforme de los vectores que generan las imágenes es importante para no generar un análisis parcial del sesgo de StyleGAN3 y DeepFace. Para realizar la representación de manera gráfica y debido a la gran dimensionalidad del espacio latente de StyleGAN3, se buscó representar el espacio de 512 dimensiones en 2 dimensiones.

En un principio se buscó realizar la reducción de dimensionalidad con la técnica t-distributed Stochastic Neighbor Embedding (t-SNE). Sin embargo en la documentación de [SkLearn Documentation \(2024\)](#), biblioteca de python que implementa t-SNE, se recomienda no utilizar la técnica de t-SNE en casos en que los datos contengan muchas dimensiones. Se exploraron técnicas alternativas de reducción de dimensionalidad y decidió finalmente utilizar el algoritmo Uniform Manifold Approximation & Projection (UMAP). UMAP es una técnica de reducción de dimensionalidad no lineal que trabaja de manera similar a t-SNE. UMAP presenta un mejor rendimiento en comparación con t-SNE cuando la dimensionalidad de los datos de entrada son muy numerosos, como se detalla en [UMAP Documentation \(2024\)](#). Los vectores se categorizaron según las características atribuidas por DeepFace de raza, género y edad.

### 5.1.3. Metodología para el análisis de sesgo en la generación y evaluación de rostros

En esta subsección se describen las metodologías para el análisis del sesgo combinado en el proceso de generación de rostros por StyleGAN3 y su posterior evaluación por DeepFace. Posteriormente, se detalla la metodología para el análisis del sesgo del entrenamiento de StyleGAN3. Finalmente se explica la metodología para el análisis de sesgo por parte de DeepFace. Se buscó analizar si el sesgo proviene del entrenamiento de StyleGAN3, del análisis provisto por DeepFace o una combinación de ambos.

## Sesgo del pipeline

Para realizar el análisis de sesgo presente en StyleGAN3 y Facebook DeepFace en conjunto, se utilizaron las imágenes generadas por StyleGAN3 presentadas en la [Subsección 5.1.1](#), para ser analizadas por Facebook DeepFace. Se buscó determinar si existe un sesgo significativo en el uso combinado de StyleGAN3 y DeepFace.

Se realizó un análisis sobre los resultados retornados por DeepFace respecto a los rostros detectados. Se comparó la proporción de rostros caracterizados con cada raza dominante, género dominante y rango etario. En la [Tabla 5.1](#) se presenta la división de edades en rangos etarios utilizada en el análisis.

<i>Rango etario</i>	<i>Rango(años)</i>
Infante	[0, 4]
Niño/a	[5, 11]
Adolescente	[12, 17]
Joven	[18, 26]
Adulto	[27, 59]
Adulto mayor	[60, $\infty$ ]

Tabla 5.1: Asignación de rangos etarios para el análisis de sesgo

Se analiza también el promedio de probabilidad de un rostro de pertenecer a una raza o género, independientemente del rasgo dominante.

## Sesgo de StyleGAN3

Para evaluar la posible presencia de sesgo en el modelo de StyleGAN3 que se utilizó para generar imágenes, se procedió a recopilar las imágenes que se emplearon en su fase de entrenamiento, obtenidas del dataset Flickr-Faces-HQ. El siguiente paso fue realizar un análisis exhaustivo de las imágenes obtenidas utilizando DeepFace, generando nuevamente histogramas para ver la distribución de raza, edad y género. Fue considerada al hacer este análisis la posibilidad de que DeepFace presente un sesgo y esto comprometa los resultados. Sin embargo, no se encontró una mejor alternativa para observar el comportamiento de StyleGAN3 de manera individual.

El objetivo del análisis fue determinar si el modelo aprendió de manera desproporcionada de ciertos grupos o características, lo que podría llevar a la generación de imágenes sesgadas.

## Sesgo de DeepFace

Para investigar la posible existencia de sesgo en DeepFace, se llevó a cabo un proceso de análisis utilizando la base de datos UTKFace ([Zhang y cols., 2017](#)) de imágenes etiquetadas. UTKFace proporciona una representación de individuos con etiquetas que abarcan atributos como género, edad y raza, además de otros que no fueron considerados para el proceso de análisis.

El proceso de análisis consistió en analizar cada imagen en la base de datos UTKFace con DeepFace y comparar los resultados generados con las etiquetas del dataset. Se buscó identificar si el sistema proporciona resultados consistentes en las diferentes categorías o si muestra alguna tendencia a cometer errores en grupos específicos. Específicamente se analizó para cada género, raza y rango etario el número de veces en los que la clasificación de DeepFace coincidió con el etiquetado de las imágenes.

### 5.1.4. Elección de rostros para la experimentación

La elección de los rostros utilizados en la configuración paramétrica y la validación se realizó teniendo en cuenta la diversidad de género, raza y edad. Se eligieron un total de nueve imágenes, de las cuales cinco son del género femenino y cuatro del género masculino.

Dentro de los rostros de género femenino se encuentra una persona andrógina, con motivo de analizar el comportamiento del algoritmo en rostros que el ojo humano tiene dificultades para definir como hombre o mujer. Se eligieron tres personas de tercera edad, tres jóvenes y tres adultos. No se eligieron niños o niñas dado que se consideró que es difícil definir el género de personas de ese grupo etario, incluso para el ojo humano.

En cuanto a razas, se encuentran entre las imágenes elegidas cuatro personas de raza blanca, dos personas de raza negra, una persona de raza india con accesorios característicos de la cultura, una persona de raza asiática y una persona de raza medio-oriental. En la [Figura 5.1](#) se presentan los rostros elegidos con el porcentaje de expansión aplicado (como se detalla en la [Subsección 5.1.5](#)) y en la [Tabla 5.2](#) se presentan las características de nombre, género y raza de cada uno.



Figura 5.1: Rostros utilizados para la experimentación. De arriba a abajo y de izquierda a derecha: I1-Margot Robbie, I2-Hombre de medio oriente, I3-Steve Harvey, I4-Mujer india, I5-Modelo desfile Hermés 2018, I6-James Charles, I7-Whoopi Goldberg, I8-Marcos Mundstock, I9-Lana Condor

<i>Instancia</i>	<i>Nombre</i>	<i>Género</i>	<i>Raza</i>
I1	Margot Robbie	Femenino	Blanca
I2	Hombre de medio oriente	Masculino	Medio oriente
I3	Steve Harvey	Masculino	Negra
I4	Mujer india	Femenino	India
I5	Modelo desfile Hermés 2018	Femenino	Blanca
I6	James Charles	Masculino	Blanco
I7	Whoopi Goldberg	Femenino	Negra
I8	Marcos Mundstock	Masculino	Blanca
I9	Lana Condor	Femenino	Asiatica

Tabla 5.2: Características de las instancias

### 5.1.5. Metodología para la elección del porcentaje de expansión de imágenes

RetinaFace realiza un recorte y centrado de los rostros reconocidos como se detalla en la [Sección 4.3.2](#). El recorte de rostros predeterminado es bastante ajustado y causa que se omitan ciertas características, como el largo del cabello o de la barba. Se analizó si un recorte de imágenes con un mayor porcentaje de expansión podría resultar en mejores aproximaciones de género y raza en los rostros.

Se llevó a cabo una serie de experimentos en el proceso de recorte y análisis de imágenes, utilizando diferentes porcentajes de expansión (0 %, 20 %, 30 %, 40 %, 50 %) para determinar cuál proporcionaba los mejores resultados. La expansión se refiere a la cantidad de área adicional alrededor del rostro que se incluye en el recorte final que será analizado por DeepFace. Para cada porcentaje de expansión utilizado se realizó un análisis detallado sobre la concordancia entre los resultados de DeepFace respecto a la raza y el género y los valores reales de las imágenes recortadas. El análisis para la elección del porcentaje de expansión permitió evaluar cómo afectan los diferentes niveles de expansión a la extracción y análisis de los rostros, asegurando así una representación más precisa de los atributos analizados en la [Subsección 5.1.6](#) y la [Subsección 5.1.8](#).

Para cada instancia se analizó con DeepFace la raza y el género. Se realizó un test de rangos para determinar cual era el mejor porcentaje de expansión. DeepFace devuelve valores más altos a los géneros o razas a los que tiene mas confianza que pertenezca el rostro, y valores más bajos cuando tiene mayor confianza de que la raza o el género no corresponde al rostro. Por lo tanto, para determinar el rango de cada expansión se buscó el resultado con valor máximo para el género o la raza a la que la instancia pertenece, y el resultado con valor mínimo para el género y la raza a los que no pertenece.

### 5.1.6. Metodología para la elección de la configuración paramétrica

Para definir la configuración paramétrica a utilizar en el algoritmo evolutivo se estudió el tamaño de la población, la probabilidad de mutación y la probabilidad de cruzamiento. En particular se utilizaron los rostros de una mujer joven y andrógina de raza blanca (I5), un hombre de tercera edad y raza negra (I3), y una mujer joven de raza asiática (I9) para cada configuración. Para cada imagen objetivo se buscó variar el género y la raza objetivo para obtener distintas configuraciones y cubrir todas las razas y géneros consideradas por el analizador. Se presentan en la [Tabla 5.3](#) las configuraciones utilizadas. Se varió además el tamaño de la población, la probabilidad de cruzamiento y la probabilidad de mutación, y se generó un total de 8 posibles configuraciones para cada combinación de rostro, género y raza objetivo. La [Tabla 5.4](#) presenta el tamaño de la población (MU), la probabilidad de cruzamiento (CXPB) y la probabilidad de mutación (MXPB) de las 8 configuraciones.

Se calculó que, en el caso de que la probabilidad de mutación tuviera un valor de 0.01, la probabilidad de que ningún alelo en un individuo mute sería aproximadamente  $0,99^{512} = 5,8e^{-3}$ . Se decidió agregar una probabilidad extra de 0.75 de que un individuo sea sujeto a un intento de mutación, con el fin de evitar que todos los individuos muten en cada generación, perdiendo potencialmente buenos individuos. Cuando la probabilidad de mutación es igual a 0.001, la probabilidad de que un individuo resulte inalterado es de  $0,999^{512} = 0,60$ , por lo que se decidió no agregar la probabilidad extra.

El análisis de similitud, al igual que los análisis de raza y género, se realizó en batches de 10 imágenes, con el fin de acelerar el tiempo total de ejecución del algoritmo.

<i>Instancia</i>	<i>Rostro objetivo</i>	<i>Género objetivo</i>	<i>Raza objetivo</i>
$I5_1$	I5	Masculino	Negra
$I5_2$	I5	Masculino	Medio-oriente
$I3_1$	I3	Femenino	India
$I3_2$	I3	Femenino	Asiática
$I9_1$	I9	Masculino	Blanca
$I9_2$	I9	Masculino	Latino-hispana

Tabla 5.3: Configuraciones de rostro, género y raza objetivo utilizadas en la configuración paramétrica

<i>Configuración</i>	<i>MU</i>	<i>CXPB</i>	<i>MXPB</i>
C1	100	0,80	0,001
C2	100	0,80	0,010
C3	100	0,65	0,001
C4	100	0,65	0,010
C5	80	0,80	0,001
C6	80	0,80	0,010
C7	80	0,65	0,001
C8	80	0,65	0,010

Tabla 5.4: Configuraciones de tamaño de población, probabilidad de cruzamiento y probabilidad de mutación utilizadas en la configuración paramétrica

### **Análisis de la configuración paramétrica**

Se evaluaron seis instancias diferentes, y para cada una de estas instancias se evaluaron ocho configuraciones diferentes. Cada configuración se ejecutó 20 veces. Para evaluar el rendimiento de cada ejecución del algoritmo, se utilizó la métrica de hipervolumen. El hipervolumen es una medida comúnmente utilizada

en optimización multi-objetivo para cuantificar el volumen del espacio dominado por un conjunto de soluciones en comparación con un punto de referencia. Se utilizó particularmente la métrica de hipervolumen relativo que compara el hipervolumen de un conjunto de soluciones con el hipervolumen del conjunto de soluciones óptimas para el problema. Considerando que no se conocía el conjunto de soluciones óptimas, se debió aproximar a través de las soluciones resultantes de las ejecuciones de cada instancia.

Primero, se recolectó y guardó todos los valores de fitness de cada una de las ejecuciones agrupadas por instancia. Luego se utilizó una función que recibe los valores de fitness y determina el conjunto de soluciones no dominadas (frente de Pareto) y el conjunto de soluciones dominadas. Los puntos del frente de Pareto representan las soluciones más eficientes en términos de los objetivos de optimización. Un punto o solución se considera no dominado o Pareto óptimo si no existe otro punto que lo supere en al menos uno de los objetivos sin empeorar en otros. Los puntos dominados son aquellos para los cuales existe al menos otra solución que es mejor en al menos un objetivo sin ser peor en ninguno de los otros.

El proceso de identificar puntos no dominados y dominados, conocido como evaluación de Pareto, permite comparar el rendimiento de las diferentes configuraciones paramétricas del algoritmo de manera efectiva. Se definió una aproximación al verdadero frente de Pareto como el frente de Pareto calculado a partir de las soluciones de todas las ejecuciones de una instancia.

A partir de la aproximación al verdadero frente de Pareto, se definió un punto de referencia ligeramente mayor al punto máximo de los valores de fitness hallados en cada dimensión. Además se calculó, utilizando el punto de referencia y la aproximación al verdadero frente de Pareto, un hipervolumen que se utilizó como referencia para cada configuración paramétrica. Posteriormente, se calculó el hipervolumen del frente de Pareto para cada configuración paramétrica utilizando los puntos no dominados de cada ejecución y el punto de referencia definido. El hipervolumen de cada frente de Pareto se comparó con el hipervolumen de referencia para obtener un hipervolumen relativo, permitiendo así comparar diferentes configuraciones en términos relativos.

### **Test de Kolmogorov-Smirnov**

Dado que los algoritmos evolutivos no son deterministas, los resultados varían entre ejecuciones independientes. Es necesario realizar una evaluación estadística para analizar los resultados obtenidos de las distintas ejecuciones. Para este caso, se evaluó la muestra del hipervolumen relativo de cada una de las 20 ejecuciones de cada instancia utilizando la prueba de Kolmogorov-Smirnov. La prueba de Kolmogorov-Smirnov compara la distribución acumulada de la muestra de una variable con una distribución determinada, en este caso la distribución normal ( $N(\mu, \sigma)$ ).

La prueba de normalidad es fundamental para determinar si se va a utilizar la media o mediana al momento de reportar los resultados. Si la distribución de los valores de la muestra obtenidas en cada ejecución sigue una distribución

normal ( $N(\mu, \sigma)$ ), entonces podemos afirmar que aproximadamente el 95 % de la distribución se encuentra en el intervalo de confianza, dado que para este caso se toma  $\alpha = 0,05$ .

### Test de rangos de Friedman

Una vez calculados los hipervolúmenes relativos de los frentes de Pareto generados para cada configuración en cada instancia, se procedió a seleccionar la configuración más adecuada para cada instancia. Se utilizó el test de rangos de Friedman, un test estadístico usado comúnmente para determinar el rango promedio de distintos algoritmos o configuraciones paramétricas al ejecutarse sobre las mismas instancias. El test de rangos de Friedman realiza primero un test de hipótesis con la hipótesis nula  $H_0$ . Que se cumpla  $H_0$  significa que no hay diferencia significativa entre los datos de hipervolúmenes sobre una misma instancia, por lo tanto todas las configuraciones siguen la misma distribución. Si se obtiene un p-valor menor a un alpha definido (en este caso  $\alpha = 0,05$ ) entonces se rechaza la hipótesis y se realiza un test post-hoc para determinar el rango promedio de cada configuración en la instancia. Dado que en el test se consideran mejores los valores más cercanos a cero, y que el hipervolumen relativo es mejor al acercarse a 1, se invirtieron los resultados de los rangos promedio para definir el rango real de cada configuración. Se definió la mejor configuración como la configuración con el menor rango real.

El test de rangos de Friedman permitió identificar la configuración paramétrica más efectiva y comprender las variaciones en el rendimiento del algoritmo. Garantiza que la configuración seleccionada sea la más adecuada para optimizar el rendimiento del algoritmo en diferentes instancias.

#### 5.1.7. Metodología para el cálculo experimental del umbral

Como se explica en la [Sección 4.3.1](#), al analizar un rostro sintético producido por StyleGan3, se genera un vector para cada uno de los objetivos. Con el fin de determinar si la distancia entre los vectores generados y cada uno de los vectores objetivo es lo suficientemente pequeña como para engañar al analizador, se debe determinar un umbral. El umbral representa la máxima distancia posible con la cual se considera que dos rostros tienen la misma raza, género o pertenecen a la misma persona. Si la distancia entre los dos vectores es mayor al umbral establecido, se da por hecho que el analizador no ha sido engañado.

Debido a las modificaciones realizadas en la capa de salida, como se explica en la [Subsección 4.3.3](#), los umbrales determinados por la biblioteca DeepFace dejan de ser precisos. Es necesario determinar un umbral acorde a las transformaciones realizadas, ya que un umbral demasiado alto genera que imágenes de características objetivo diferentes sean identificadas como si las compartieran. Por otro lado, un umbral demasiado bajo genera que rostros que comparten características objetivo sean descartados como si no las compartieran.

Como las transformaciones realizadas y los modelos utilizados para cada análisis son diferentes para cada objetivo, los umbrales deben de ser determinados independientemente. Para determinar el umbral de similitud se utilizaron imágenes tomadas de la base de Labeled Faces in the Wild (LFW) (Huang y cols., 2007) que se encuentran etiquetadas con el nombre de la persona a la que pertenecen. Con estas imágenes se realizaron 4000 comparaciones de diferentes combinaciones de rostros de las mismas personas. En cuanto al género y la raza, las imágenes para la comparación se tomaron de la base FairFace, y de igual manera que con la similitud, se realizaron 4000 comparaciones de imágenes con sus respectivas razas y géneros.

### 5.1.8. Metodología para la validación

La [Tabla 5.5](#) detalla las combinaciones de rostro, raza y género objetivos utilizadas en el proceso de validación. Luego de terminada la etapa de configuración paramétrica y definidos los hiperparámetros a utilizar en la validación, se construyó para cada una de las instancias de validación una aproximación al verdadero frente de Pareto combinando todas las soluciones no dominadas halladas en 60 ejecuciones del algoritmo.

<i>Instancia</i>	<i>Persona objetivo</i>	<i>Raza objetivo</i>	<i>Género objetivo</i>
I1	Margot Robbie	Asiática	Masculino
I2	Hombre de medio oriente	Negra	Femenino
I4	Mujer india	Blanca	Masculino
I6	James Charles	Medio oriente	Femenino
I7	Whoopi Goldberg	Latina hispana	Masculino
I8	Marcos Mundstock	India	Femenino

Tabla 5.5: Instancias con personas, raza, y género objetivo

Una vez terminadas las 60 ejecuciones, se obtuvo el hipervolumen de la aproximación del verdadero frente de Pareto para cada instancia. A continuación se tomaron las últimas 20 ejecuciones del algoritmo para cada una de las instancias, y se registró el hipervolumen relativo y las soluciones no dominadas de cada generación para poder observar la evolución de los mismos.

Para analizar la convergencia del algoritmo, se tomaron los hipervolumenes relativos de las 20 ejecuciones de cada instancia para calcular la media, la desviación estándar y el valor máximo. El hipervolumen relativo se calculó en comparación con el hipervolumen obtenido a partir de la aproximación al verdadero frente de Pareto obtenido anteriormente. También se calculó la media y la desviación estándar de la distancia generacional invertida y el spacing de las soluciones.

El spacing es una métrica utilizada para evaluar la distribución de los puntos no dominados obtenidos como solución del problema multi-objetivo. El spacing mide cuán uniformemente distribuidos están estos puntos a lo largo del frente

de Pareto. Un valor ideal de cero en la métrica de spacing indica que todos los puntos no dominados obtenidos por el algoritmo están equiespaciados, dado que la distancia entre cada punto y sus vecinos es constante. Un valor de spacing cercano a cero sugiere una buena distribución de las soluciones, mientras que valores más altos indican que las soluciones están distribuidas de manera desigual, con algunos puntos más concentrados y otros más dispersos. El spacing es importante para asegurar que las soluciones obtenidas cubren de manera uniforme el frente de Pareto, proporcionando una representación completa y equitativa de las posibles soluciones del problema (Deb, 2001).

La distancia generacional invertida (IGD) es una métrica utilizada para evaluar la proximidad de las soluciones no dominadas obtenidas con respecto a los elementos del verdadero frente de Pareto. La distancia generacional mide qué tan lejos están los elementos obtenidos del verdadero frente de Pareto y la distancia generacional invertida, evalúa qué tan cerca están los elementos del verdadero frente de Pareto de las soluciones no dominadas obtenidas por el algoritmo. Busca proporcionar una medida de cuán bien las soluciones generadas representan el verdadero frente de Pareto. Un valor bajo indica que las soluciones generadas están muy cerca del verdadero frente de Pareto, lo que sugiere una alta calidad en las soluciones obtenidas. Por el contrario, un valor alto indicaría que las soluciones generadas están lejos del verdadero frente de Pareto, lo que sugiere una menor calidad (Deb, 2001).

### 5.1.9. Ambiente de desarrollo y ejecución

Los experimentos fueron llevados a cabo en la plataforma de cómputo de alto rendimiento ClusterUY, en procesadores Xeon Gold 6138 y unidades de procesamiento gráfico (GPU) NVIDIA P100. El uso de GPUs es necesario para acelerar de manera significativa las tareas de generación y reconocimiento de imágenes.

El proyecto en su totalidad fue desarrollado en Python 3.11.4 y se utilizó la biblioteca DEAP versión 1.4.1 para la implementación del algoritmo evolutivo. La versión de DeepFace utilizada para la detección y análisis facial fue la 0.0.84. Para asegurar la correcta gestión de las bibliotecas y dependencias requeridas por el proyecto, entre ellas DEAP, PyTorch y TensorFlow, se utilizó un ambiente conda. El entorno de gestión de paquetes y ambientes virtuales de conda facilitó la instalación y el manejo de las herramientas necesarias.

## 5.2. Resultados del análisis experimental

En esta sección se detallan los resultados del análisis experimental para el algoritmo evolutivo desarrollado, así como los hallazgos sobre el desempeño de los modelos de detección, la exploración del espacio de vectores de entrada de StyleGAN3, y el sesgo en la generación y evaluación de rostros. También se presentan los resultados relacionados con la elección del porcentaje de expansión de imágenes, la configuración paramétrica, el cálculo experimental del umbral, y la validación de los métodos utilizados.

### 5.2.1. Resultados del análisis de desempeño de los modelos de detección y análisis

A partir de las 700.000 imágenes generadas por StyleGAN3, se notó que alrededor de 630.000 fueron reconocidas como rostros y analizadas por DeepFace con OpenCV. Debido a que cerca del 10% del total de imágenes generadas no fueron reconocidas como rostros, se realizó un análisis sobre los rostros no reconocidos. Se encontró que a pesar de que las imágenes no eran reconocidas como rostros por OpenCV, muchas eran reconocibles como rostros a simple vista como se muestra en la [Figura 5.2](#), por lo que se comenzó una investigación sobre las otras opciones de reconocedores disponibles en la biblioteca DeepFace.



Figura 5.2: Rostros no detectados por OpenCV que resultan reconocibles a simple vista

La [Tabla 5.6](#) reporta el tiempo total ( $T$ ), el tiempo promedio por imagen ( $\bar{T}_i$ ) y los errores de detección ( $E$ ) para cada modelo de detección ofrecido por DeepFace para el análisis de 1000 imágenes tomadas de ffhq.

Se realizó nuevamente el reconocimiento y análisis para las 700.000 imágenes presentadas en la [Subsección 5.1.1](#), esta vez utilizando los cuatro mejores detectores presentados en la [Tabla 5.6](#). Se muestran en la [Tabla 5.7](#) el número de imágenes no reconocidas como rostros ( $\#I_{nr}$ ), el porcentaje de fallas ( $P_f$ ) y el porcentaje de mejora con respecto a OpenCV ( $P_m$ ). En el caso de RetinaFace el porcentaje de mejora con respecto al detector original OpenCV mejoró significativamente.

<i>Modelo</i>	$T$	$\bar{T}_i$	$E$
RetinaFace	208	0,21	0
Mediapipe	772	0,85	94
Yunet	106	0,12	80
SSD	367	0,36	0
Fastmtcnn	370	0,37	1000

Tabla 5.6: Comparación de modelos de detección de DeepFace

<i>Detector</i>	$\#I_{nr}$	$P_f$	$P_m$
RetinaFace	23852	3,41 %	193,48 %
SSD	43877	6,27 %	59,54 %
Yunet	55102	7,87 %	27,04 %
MediaPipe	57837	8,26 %	21,03 %

Tabla 5.7: Comparación de mejores detectores de DeepFace

Debido a que el tiempo de ejecución de RetinaFace no fue significativamente mayor al de SSD, Yunet y MediaPipe, y dado que el número de imágenes no reconocidas como rostros fue menor al del resto de las implementaciones, se optó por usar RetinaFace como reconocedor para el proyecto.

Se regeneraron además algunas imágenes en las que RetinaFace no detectó rostros, con el fin de evaluar visualmente si no presentaban ningún rostro. Al analizar las imágenes se identificó que efectivamente no existían rostros reconocibles o se encontraban extremadamente distorsionados. Algunas de las imágenes regeneradas se presentan en la [Figura 5.3](#).

La [Tabla 5.8](#) muestra el Tiempo total de ejecución ( $T$ ) y el Tiempo promedio por imagen ( $\bar{T}$ ) de cada modelo de análisis provisto por la biblioteca DeepFace.

<i>Modelo</i>	$T$	$\bar{T}$
VGG-Face	180	0,180
Facenet	527	0,527
Facenet512	582	0,581
OpenFace	298	0,298
DeepFace	3722	3,718
DeepID	104	0,104
ArcFace	249	0,249

Tabla 5.8: Comparación de analizadores

Se concluyó que DeepFace tiene un mayor tiempo de ejecución total y promedio, mientras que otros analizadores, como VGG-Face y DeepID, mostraron tiempos de ejecución significativamente menores. Los resultados presentados en



Figura 5.3: Imágenes no reconocidas como rostros por RetinaFace

[Taigman y cols. \(2014\)](#) detallan la alta precisión de DeepFace tras analizar correctamente un 97.35% de las imágenes presentadas en LFW ([Huang y cols., 2007](#)). Debido a los resultados de [Taigman y cols. \(2014\)](#) y continuando con lo presentado por [Correa y cols. \(2023\)](#), se decidió usar DeepFace a pesar de la diferencia en los tiempos de ejecución.

### 5.2.2. Resultados del análisis de la exploración del espacio de vectores de entrada de StyleGAN3

Las figuras [5.4](#), [5.5](#) y [5.6](#) presentan el resultado de realizar la reducción de dimensionalidad de los vectores de las imágenes generadas en la [Subsección 5.1.1](#) utilizando UMAP. La [Figura 5.4](#) muestra los vectores generados categorizados por la raza, donde cada color representa una raza. Las figuras [5.5](#) y [5.6](#) muestran los vectores categorizados por género donde la [Figura 5.5](#) muestra los vectores categorizados como hombre y la [Figura 5.6](#) los vectores categorizados como mujer.

Se observó en las figuras [5.4](#), [5.5](#) y [5.6](#) que los vectores se encuentran uniformemente distribuidos en el espacio y que las clases no se agrupan ni forman clusters. La falta de clusters tiene sentido debido a la alta reducción de dimensionalidad que se realiza. No se pudieron identificar regiones donde alguna raza o género resultara predominante, lo que hubiera resultado útil para definir una heurística al inicializar una población.

La [Figura 5.7](#) presenta el resultado de graficar, usando también UMAP, el conjunto de las imágenes generadas, clasificándolas entre las identificadas como rostros por RetinaFace y las que generaron un error en la detección rostros.

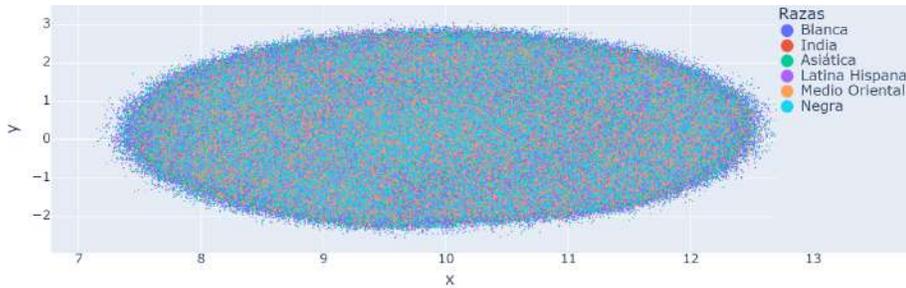


Figura 5.4: Análisis de Espacio Latente con UMAP para la raza

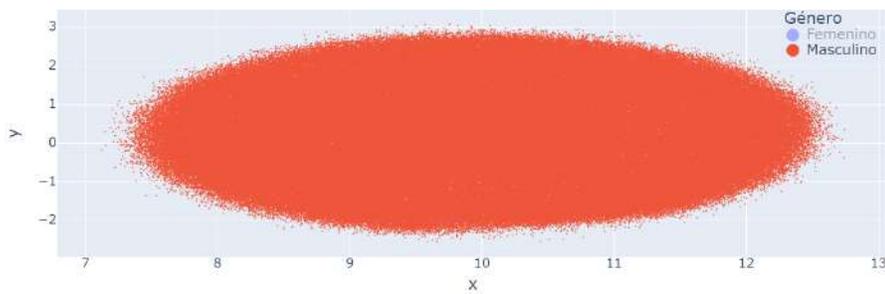


Figura 5.5: Análisis de Espacio Latente con UMAP para el género masculino

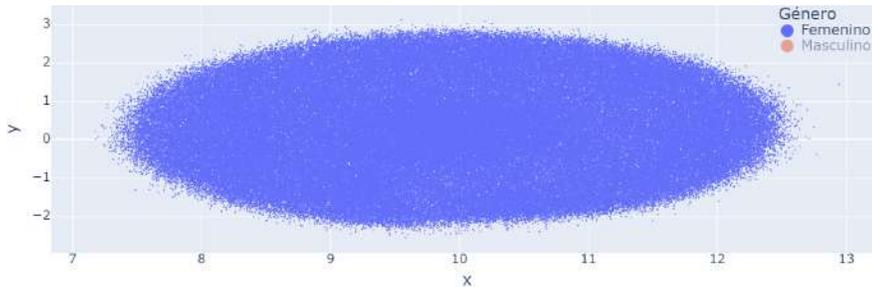


Figura 5.6: Análisis de Espacio Latente con UMAP para el género femenino

En la [Figura 5.7](#) existe una zona donde el algoritmo de RetinaFace reconoce más rostros generados por StyleGAN3. Los vectores del extremo medio y superior izquierdo corresponden a caras fácilmente identificables para el detector. Analizar posibles causas de que sean fácilmente identificables puede ser útil en futuras investigaciones.

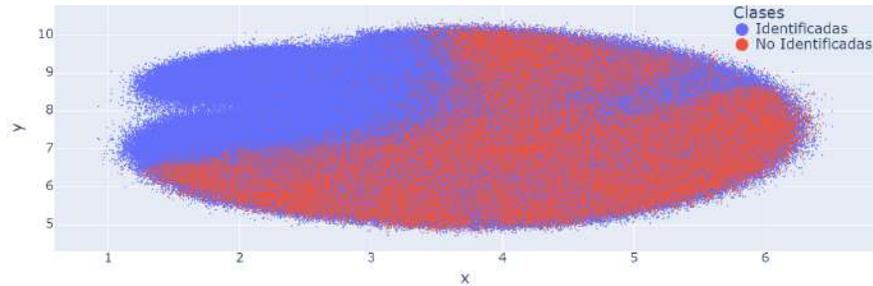


Figura 5.7: Análisis de detección de rostros con UMAP

### 5.2.3. Resultados del análisis de sesgo en la generación y evaluación de rostros

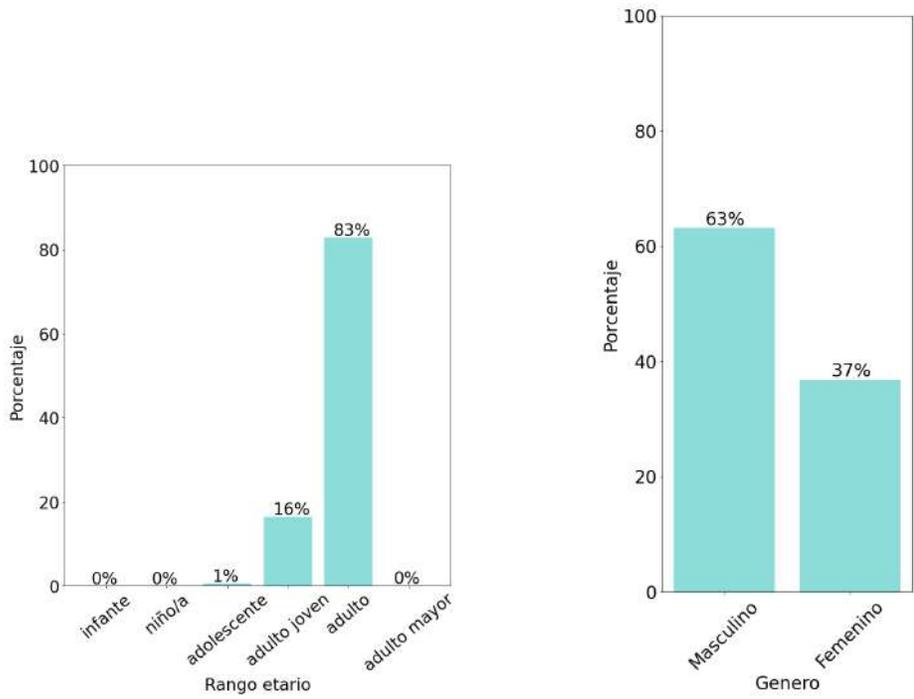
En esta subsección se describen los resultados del el análisis del sesgo combinado en el proceso de generación de rostros por StyleGAN3 y su posterior evaluación por DeepFace. Posteriormente, se presentan los resultados del análisis de sesgo del entrenamiento de StyleGAN3 y el análisis de sesgo de DeepFace.

#### Sesgo del pipeline

La [Figura 5.8](#) presenta la distribución de edad, raza y género de los rostros analizados por DeepFace en las imágenes generadas en la [Subsección 5.1.1](#). A partir de la [Figura 5.8](#) se concluyó que el pipeline está altamente sesgado hacia la representación de rostros de adultos de género masculino y de raza blanca en comparación con las otras etnicidades y grupos etarios.

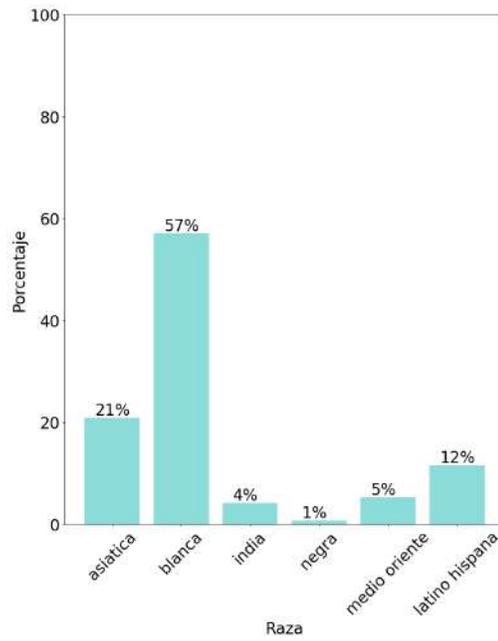
Del total de las imágenes reconocidas, el 57% fueron clasificadas como personas de raza blanca, el 0.79% fueron identificadas como personas de raza india y el 4.16% como personas de raza negra. En cuanto a las distribuciones de edades, el 82.9% de las imágenes fueron clasificadas por DeepFace como adultos y menos de 1% de rostros identificados como adultos mayores, niños e infantes. Por último, en cuanto al género de las imágenes generadas, el 63% fue clasificado como masculino y el 37% fue clasificado como femenino.

Debido a que los resultados obtenidos indicaron un claro sesgo hacia personas de género masculino, raza blanca y edad adulta, se decidió investigar si el sesgo era inherente a StyleGAN3 al generar las imágenes, a DeepFace al realizar el reconocimiento o a una combinación de ambas.



(a) Distribución de la edad

(b) Distribución del género

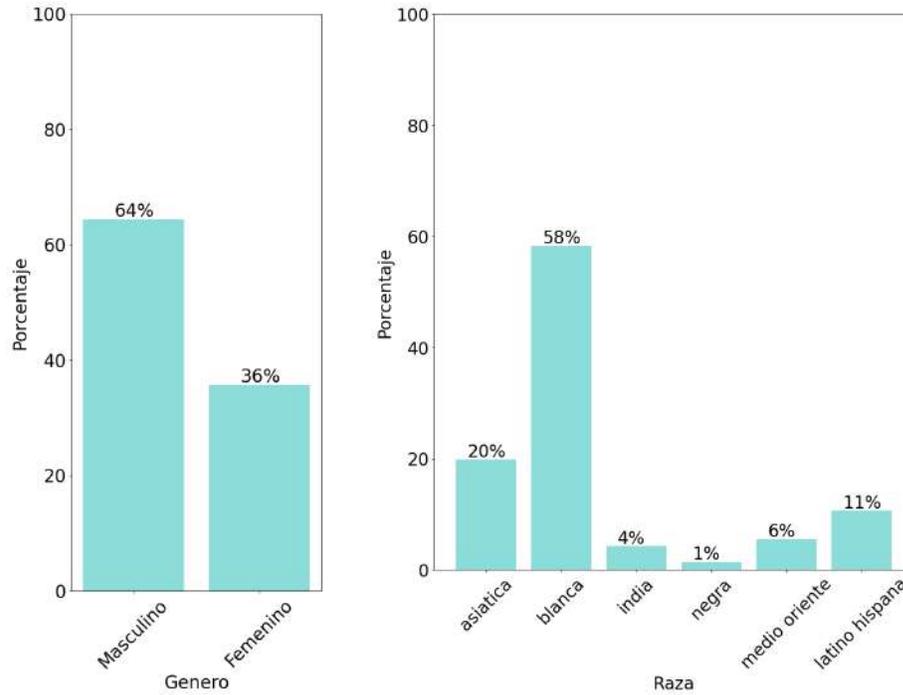


(c) Distribución de la raza

Figura 5.8: Distribución de edad, género y raza

### Sesgo de StyleGAN3

Las figuras 5.9a, 5.9b y 5.10 presentan las distribuciones de edad, raza y género de las imágenes analizadas por DeepFace, extraídas del dataset ffhq.



(a) Distribución del género según análisis con StyleGAN3

(b) Distribución de la raza según análisis con StyleGAN3

Figura 5.9: Distribución de género y raza según análisis con StyleGAN3

Las figuras 5.9a, 5.9b y 5.10 muestran que existe un sesgo similar al que se observó en el análisis del sesgo del pipeline, ya que el conjunto de imágenes con las que se entrenó el modelo generan distribuciones de raza, género y edad muy similares a las generadas [Subsección 5.1.1](#).

Sin embargo, el sesgo hacia estas cualidades en los individuos generados podría no provenir solamente de StyleGAN. La distribución despareja puede también deberse a un sesgo de DeepFace al momento de caracterizar los rostros, por lo que se realizó un análisis del funcionamiento de DeepFace.

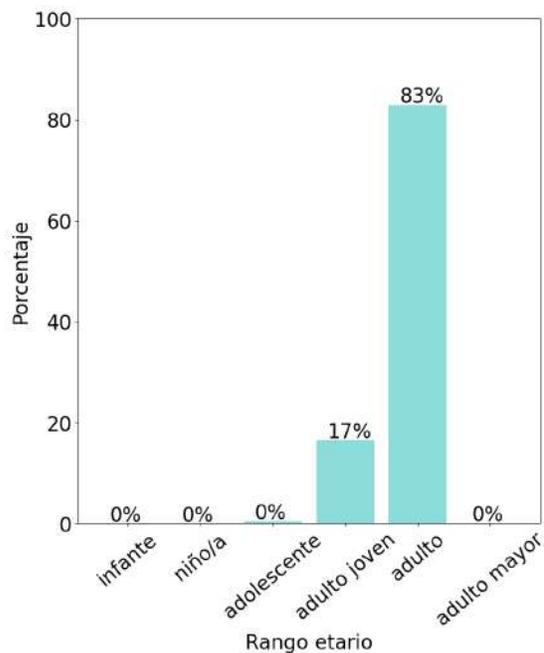


Figura 5.10: Distribución de la edad según análisis con StyleGAN3

### Sesgo de DeepFace

Las figuras 5.13, 5.11 y 5.12 presentan la distribución de los rostros según si fueron clasificados con el mismo valor que su etiqueta (verde) o fueron clasificados con otro valor (rojo) en la clasificación de edad, raza y género de los rostros analizados por DeepFace.

Debido a que las figuras 5.13, 5.12 y 5.11 mostraron un número significativo de fallos, se optó por llevar a cabo un análisis adicional. Se decidió comparar las distribuciones reales de edad, raza y género con las distribuciones extraídas del análisis de DeepFace. Las figuras 5.14, 5.15 y 5.16 presentan a la izquierda las distribuciones de edad, raza y género, según las etiquetas de las imágenes, y a la derecha las distribuciones de los resultados obtenidos por DeepFace.

Se identificó en la Figura 5.14 que la distribución de las etiquetas de género en el dataset de UTKFace es equitativa y que DeepFace tiende a sobrerrepresentar al género masculino. La sobrerrepresentación del género masculino sugiere que el problema de sesgo de género es en parte causa de DeepFace.

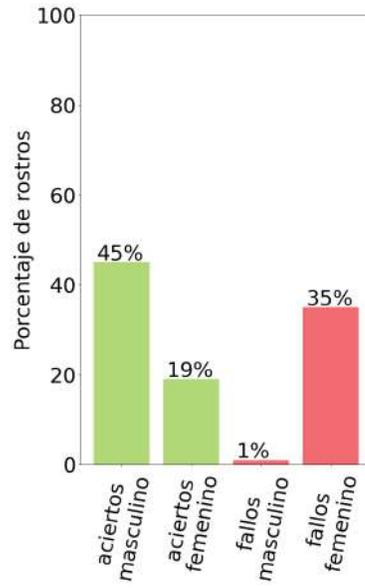


Figura 5.11: Distribución del género según análisis con DeepFace

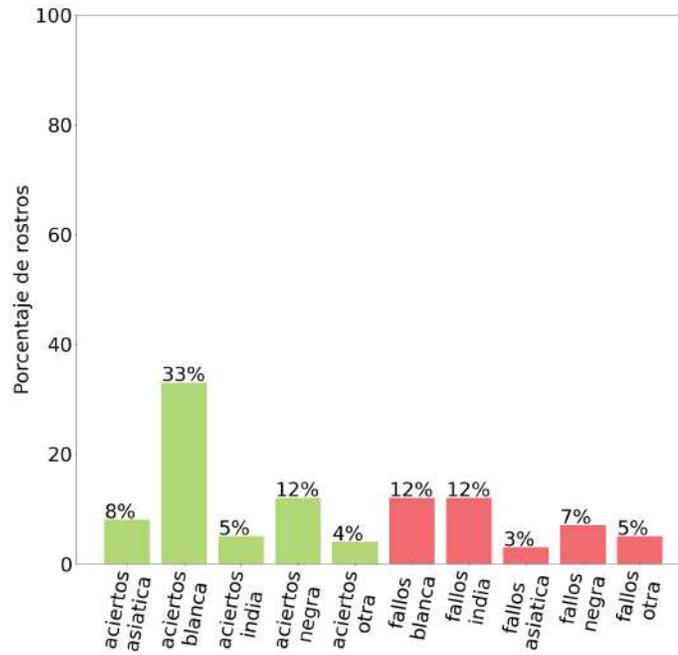


Figura 5.12: Distribución de la raza según análisis con DeepFace

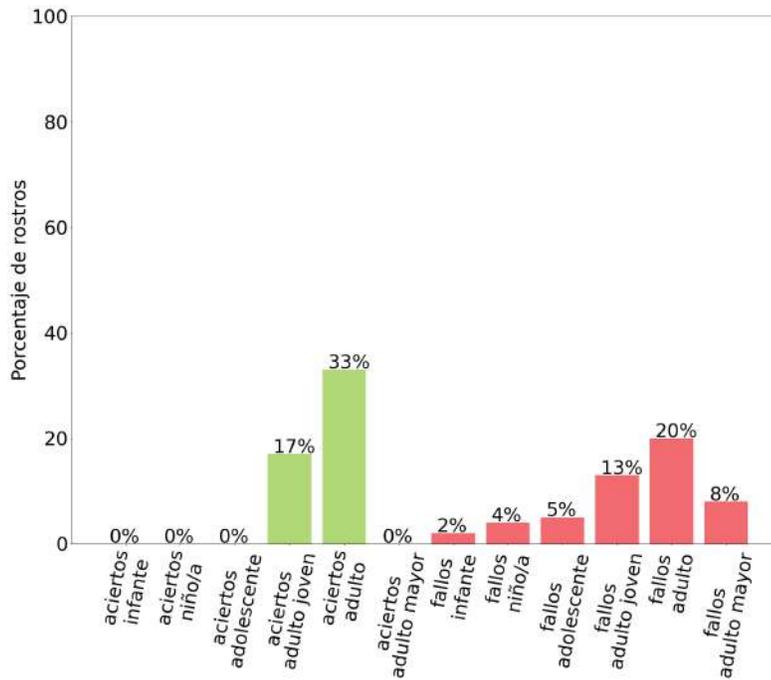
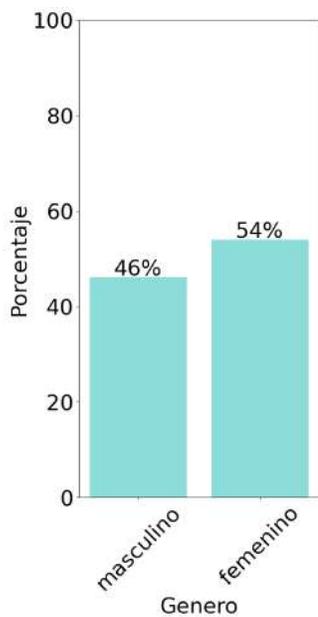


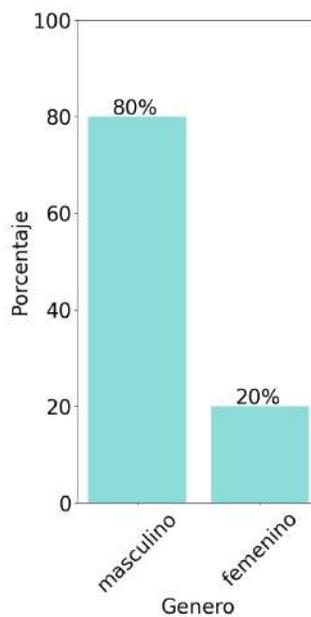
Figura 5.13: Distribución de la edad según análisis con DeepFace

La [Figura 5.16](#) detalla que DeepFace tiende a generar fallos en la clasificación para las imágenes que representan a personas que no se encuentran en la categoría de adultos o adultos jóvenes, ya que los rangos de adultos y adultos jóvenes fueron los únicos intervalos en los que DeepFace clasificó a las imágenes analizadas. Debido a los errores significativos en la representación de rangos etarios, se decidió que la característica de la edad no fuera tomada en cuenta como posible atributo a evaluar en la función de fitness.

Por último, si bien la [Figura 5.15](#) muestra errores en la distribución racial, principalmente en personas de origen indio, las distribuciones de raza son mayoritariamente similares entre el análisis de DeepFace y las etiquetas de las imágenes.

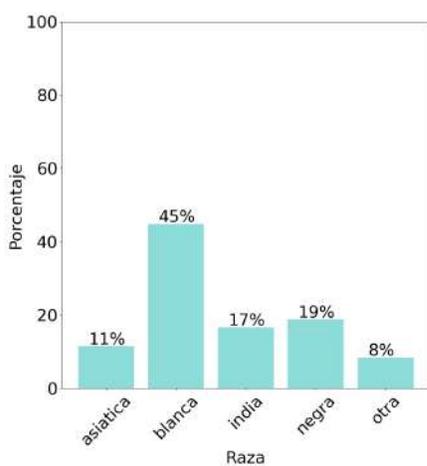


(a) Distribución de género de las etiquetas de UTKFace

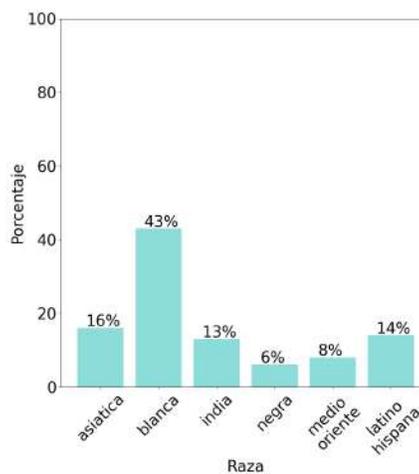


(b) Distribución de género del análisis de DeepFace

Figura 5.14: Comparación en análisis de género

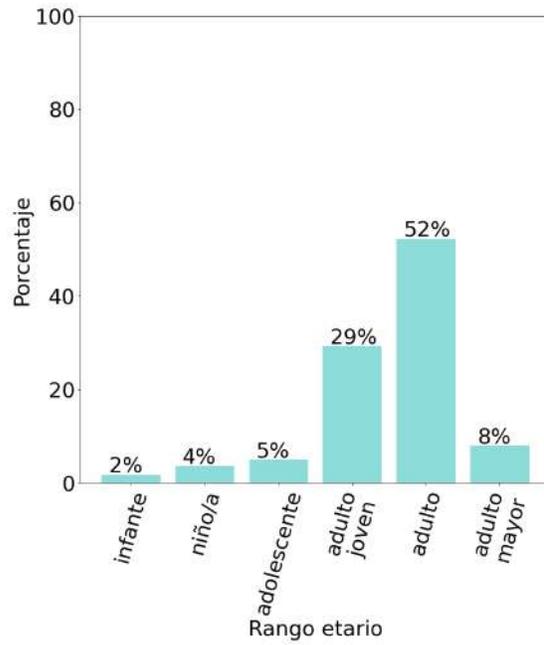


(a) Distribución de raza de las etiquetas de UTKFace

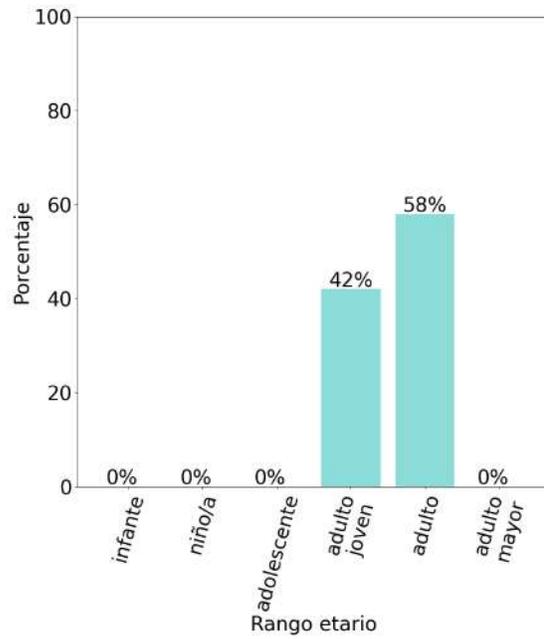


(b) Distribución de raza del análisis de DeepFace

Figura 5.15: Comparación en análisis de raza



(a) Distribución de edad de las etiquetas de UTK-Face



(b) Distribución de edad del análisis de DeepFace

Figura 5.16: Comparación en análisis de edad

### 5.2.4. Resultados de la elección del porcentaje de expansión de imágenes

Para las instancias presentadas en la [Tabla 5.2](#) se realizó una evaluación comparativa de los valores obtenidos con diferentes porcentajes de expansión. La [Tabla 5.10](#) muestra, para cada combinación de instancia y porcentaje de expansión, los rangos obtenidos y los resultados retornados por DeepFace al analizar el género. Los porcentajes de expansión seleccionados para el análisis fueron 0% ( $E(0\%)$ ), 20% ( $E(20\%)$ ), 30% ( $E(30\%)$ ), 40% ( $E(40\%)$ ) y 50% ( $E(50\%)$ ).

Por otro lado, la [Tabla 5.11](#) muestra los rangos obtenidos y los resultados retornados por DeepFace al analizar la raza. Para cada instancia se detalla el valor de la raza a la que pertenece, la sumatoria de los valores devueltos por DeepFace para las otras razas (Otros) y el rango. En la [Sección B.1](#) se muestran las tablas de la comparación de los valores de raza desagregados.

La [Tabla 5.9](#) detalla los rangos promedio para el género ( $P_g$ ), para la raza ( $P_r$ ) y el promedio de ambos rangos ( $P_{gr}$ ) en cada uno de los porcentajes de expansión. Los rangos con valor mínimo para  $P_g$ ,  $P_r$  y  $P_{gr}$  se encuentran resaltados.

	$E(0\%)$	$E(20\%)$	$E(30\%)$	$E(40\%)$	$E(50\%)$
$P_g$	3,44	3,33	3,00	<b>2,44</b>	2,78
$P_r$	3,33	3,00	2,78	<b>2,67</b>	3,22
$P_{gr}$	3,39	3,17	2,89	<b>2,56</b>	3,00

Tabla 5.9: Rangos promedio para cada porcentaje de expansión

Debido a que los rangos mínimos corresponden en los tres casos a la expansión del 40%, se decidió utilizarla en el resto del proyecto. El recorte seleccionado demostró ser el que obtuvo mejores resultados en la categoría de género y de raza, manteniendo un equilibrio entre la fidelidad de los atributos raciales y de género.

		<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
I1	Femenino	1,36	1,72	2,08	2,46	2,87
	Masculino	-2,15	-2,79	-3,31	-3,80	-4,34
	Rango	5	4	3	2	1
I2	Femenino	-1,70	-1,29	-1,12	-0,99	-0,94
	Masculino	2,04	1,57	1,43	1,18	1,12
	Rango	1	2	3	4	5
I3	Femenino	-5,53	-5,06	-4,57	-4,40	-3,86
	Masculino	6,89	6,27	5,68	5,47	4,72
	Rango	1	2	3	4	5
I4	Femenino	0,69	0,70	0,59	1,03	1,42
	Masculino	-1,24	-1,30	-1,04	-1,82	-2,37
	Rango	4	3	5	2	1
I5	Femenino	0,87	1,40	0,85	0,55	0,70
	Masculino	-1,85	-2,42	-1,54	-0,98	-1,19
	Rango	5	3	4	1	2
I6	Femenino	2,22	-0,93	-1,60	-1,26	-1,12
	Masculino	-3,68	0,98	1,88	1,51	1,32
	Rango	5	4	1	2	3
I7	Femenino	-1,63	-1,39	-1,00	-0,81	-0,22
	Masculino	1,74	1,41	0,97	0,75	0,03
	Rango	5	4	3	2	1
I8	Femenino	-2,94	-2,87	-2,88	-2,87	-2,47
	Masculino	3,52	3,64	3,69	3,74	3,21
	Rango	4	3	2	1	5
I9	Femenino	5,53	4,67	5,07	4,91	5,53
	Masculino	-7,85	-6,66	-7,27	-7,00	-7,80
	Rango	1	5	3	4	2

Tabla 5.10: Resultados de distribución de género con diferentes porcentajes de expansión

		$E(0\%)$	$E(20\%)$	$E(30\%)$	$E(40\%)$	$E(50\%)$
I1	Blanco	1,58	2,68	2,76	3,18	3,65
	Otros	-29,93	-51,89	-56,87	-68,28	-67,93
	Rango	5	4	3	2	1
I2	Medio Oriental	-0,79	-1,06	-0,77	-0,30	-0,45
	Otros	-2,25	-1,42	-1,78	-2,07	-2,06
	Rango	3	5	4	1	2
I3	Negro	9,81	6,03	6,15	7,18	6,17
	Otros	-85,68	-88,23	-79,28	-85,16	-78,59
	Rango	1	5	4	2	3
I4	Indio	1,44	1,49	1,69	1,41	1,32
	Otros	-12,21	-12,09	-13,47	-12,42	-10,77
	Rango	3	4	1	2	5
I5	Blanco	4,56	5,05	4,83	4,52	4,50
	Otros	-91,34	-124,60	-153,00	-173,92	-156,42
	Rango	5	4	2	1	3
I6	Blanco	1,35	2,31	1,88	1,86	1,69
	Otros	-17,37	-28,85	-20,50	-22,77	-18,90
	Rango	5	1	3	2	4
I7	Negro	5,68	6,48	5,62	5,29	5,34
	Otros	-36,10	-44,98	-37,64	-35,38	-39,12
	Rango	2	1	3	5	4
I8	Blanco	1,09	1,90	1,91	1,63	1,79
	Otros	-12,30	-24,70	-21,78	-18,93	-18,24
	Rango	5	1	2	4	3
I9	Asiático	0,84	0,45	0,13	-0,23	-0,18
	Otros	-41,13	-62,62	-94,30	-112,89	-110,04
	Rango	1	2	3	5	4

Tabla 5.11: Resultados de distribución de raza con diferentes porcentajes de expansión

### 5.2.5. Resultados de la configuración paramétrica

Se calcularon los hipervolumenes relativos de las 20 ejecuciones para cada configuración paramétrica de las diferentes instancias. Las tablas en la [Subsección A.1.2](#) presentan los hipervolumenes de cada una de las ejecuciones para cada configuración paramétrica.

Se realizó el test de normalidad de Kolmogorov-Smirnov para los resultados de cada instancia, con el fin de identificar si la muestra presentaba una distribución normal. En todos los casos se obtuvo un p-valor mayor a 0.05, por lo que la hipótesis de que la distribución de cada combinación de parámetros procedía con una confianza razonable de la distribución normal fue aceptada. Debido a que los resultados obtenidos presentaron distribución normal, se definió utilizar la media y la desviación estándar como métricas para analizar la calidad de las soluciones en la etapa de validación. La [Tabla 5.12](#) presenta los p-valores obtenidos para cada combinación de instancia con las distintas configuraciones paramétricas.

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
<i>I5</i> <sub>1</sub>	0,921	0,435	0,468	0,917	0,852	0,968	0,891	0,512
<i>I5</i> <sub>2</sub>	0,899	0,749	0,777	0,937	0,796	0,240	0,826	0,897
<i>I3</i> <sub>1</sub>	0,821	0,929	0,895	0,603	0,981	0,939	0,764	0,283
<i>I3</i> <sub>2</sub>	0,732	0,878	0,554	0,999	0,325	0,699	0,938	0,635
<i>I9</i> <sub>1</sub>	0,166	0,851	0,685	0,615	0,799	0,458	0,327	0,548
<i>I9</i> <sub>2</sub>	0,964	0,593	0,730	0,669	0,432	0,956	0,653	0,995

Tabla 5.12: Resultados de p-valor de test de normalidad de Kolmogorov-Smirnov para cada instancia y configuración

La [Tabla 5.13](#) muestra la media y la desviación estándar del hipervolumen relativo para cada configuración paramétrica de cada instancia. A partir de los datos resumidos en la [Tabla 5.13](#) se realizó un test de rangos para determinar la mejor configuración paramétrica.

#### Test de rangos de Friedman

La [Tabla 5.14](#) presenta los resultados del test de rangos de Friedman, presentados con mayor detalle en la [Subsección A.1.1](#). La tabla muestra el promedio de rangos de 20 ejecuciones para cada configuración paramétrica en cada instancia, resaltando el menor valor promedio para cada instancia. La [Subsección A.1.1](#) presenta una tabla por instancia donde se suman los rangos de cada configuración a lo largo de las 20 ejecuciones. La fila  $\sum_{Rango}$ , muestra la suma para cada configuración. Luego se calcula el promedio de los rangos sumados para cada configuración, reflejándose en la fila  $P_{Rango}$ . Basado en los promedios de rangos, como se explica en la [Subsección 5.1.6](#) se invierte el rango para cada configuración y se muestra en la fila *Rango*.

		<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
<i>I5<sub>1</sub></i>	<i>Media</i>	0,702	0,703	0,704	0,748	0,670	0,696	0,670	0,699
	<i>Std</i>	0,169	0,161	0,166	0,174	0,156	0,164	0,157	0,162
<i>I5<sub>2</sub></i>	<i>Media</i>	0,593	0,615	0,631	0,659	0,572	0,608	0,593	0,654
	<i>Std</i>	0,153	0,155	0,164	0,168	0,150	0,157	0,154	0,165
<i>I3<sub>1</sub></i>	<i>Media</i>	0,653	0,682	0,666	0,698	0,637	0,661	0,618	0,671
	<i>Std</i>	0,157	0,163	0,166	0,179	0,150	0,164	0,152	0,161
<i>I3<sub>2</sub></i>	<i>Media</i>	0,723	0,744	0,732	0,750	0,658	0,707	0,656	0,699
	<i>Std</i>	0,187	0,186	0,192	0,186	0,161	0,176	0,166	0,181
<i>I9<sub>1</sub></i>	<i>Media</i>	0,608	0,639	0,616	0,677	0,589	0,624	0,596	0,610
	<i>Std</i>	0,149	0,151	0,151	0,167	0,142	0,154	0,147	0,144
<i>I9<sub>2</sub></i>	<i>Media</i>	0,721	0,697	0,698	0,735	0,676	0,710	0,671	0,713
	<i>Std</i>	0,172	0,169	0,168	0,175	0,166	0,170	0,166	0,173

Tabla 5.13: Resultados del hipervolumen relativo en la configuración perimétrica

	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
<i>I5<sub>1</sub></i>	3,45	2,95	3,25	<b>1,20</b>	5,20	3,35	5,30	3,30
<i>I5<sub>2</sub></i>	3,85	3,45	2,90	<b>2,35</b>	4,90	3,65	4,15	2,75
<i>I3<sub>1</sub></i>	3,95	2,65	3,35	<b>2,50</b>	4,45	3,60	4,60	2,90
<i>I3<sub>2</sub></i>	3,25	2,55	2,75	<b>2,45</b>	4,75	3,40	5,05	3,80
<i>I9<sub>1</sub></i>	4,00	2,25	3,65	<b>1,60</b>	4,75	3,30	4,40	4,05
<i>I9<sub>2</sub></i>	2,70	3,65	3,80	<b>2,40</b>	5,00	2,90	4,85	2,70

Tabla 5.14: Promedio de rangos para cada configuración paramétrica en cada instancia.

Los valores más bajos de rangos indican una mejor aproximación del hipervolumen generado con cada configuración a la aproximación del hipervolumen óptimo. Por otro lado, un valor de rango más alto, indica una peor aproximación del hipervolumen relativo. Debido a que la configuración C4 presentó los rangos más bajos para todas las instancias, se decidió utilizarla para la validación. Se presentan en la [Tabla 5.15](#) los valores elegidos para los hiperparámetros.

<i>Parámetro</i>	<i>Valor</i>
Cantidad de generaciones	100
Individuos en la población	100
Probabilidad de cruzamiento	0,65
Probabilidad de mutación	0,01

Tabla 5.15: Configuración de hiperparámetros del algoritmo evolutivo

### 5.2.6. Resultados del cálculo experimental del umbral

Los valores de distancia resultantes de las comparaciones presentadas en [Subsección 5.1.7](#) se utilizaron para generar las métricas de media ( $\mu$ ), desviación estándar ( $\sigma$ ) y máximo para cada objetivo en la [Tabla 5.16](#).

<i>Objetivo</i>	<i>Media (<math>\mu</math>)</i>	<i>Std (<math>\sigma</math>)</i>	<i>Máximo</i>
Similitud	0,284	0,084	0,630
Raza	0,817	0,126	1,200
Género	0,234	0,137	0,818

Tabla 5.16: Distribución de Distancias en Comparaciones de Imágenes

En base a los valores presentados en la [Tabla 5.16](#) se determinó que un posible umbral para cada valor era la suma de la media con la desviación estándar ( $\mu + \sigma$ ). Con el fin de validar el umbral seleccionado se procedió a realizar una comparación con otro set de imágenes.

Para la métrica de similitud, se tomaron imágenes etiquetadas de LFW. Esta vez, se realizaron 4000 comparaciones de pares de imágenes de la misma persona y de personas diferentes. Para la raza, se tomaron imágenes etiquetadas de FairFace, y se compararon con la raza determinada por las etiquetas y con una raza aleatoria de las restantes. Para el género, se procedió de manera similar a la raza, comparando las imágenes con el género correspondiente y con el opuesto.

Se analizó la concordancia entre los resultados predichos y las etiquetas de las bases de datos, lo que permitió elaborar una matriz de confusión. Un resumen de estos resultados se presenta en la [Tabla 5.17](#). Se contabilizaron los falsos positivos (FP), falsos negativos (FN), verdaderos positivos (TP) y verdaderos negativos (TN).

<i>Objetivo</i>	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>
Similitud	2035	898	1278	141
Raza	1811	1253	747	189
Género	1545	1191	809	455

Tabla 5.17: Resumen de las matrices de confusión del análisis del umbral  $\mu + \sigma$

En base a los resultados de la [Tabla 5.17](#) se generaron las métricas de *precision*, *recall*, *accuracy* y *f1-score* presentadas en la [Tabla 5.18](#).

Debido a la gran cantidad de falsos positivos utilizando el umbral  $\mu + \sigma$ , se decidió ejecutar nuevamente la validación, esta vez utilizando como umbral la media más media desviación estándar ( $\mu + \frac{\sigma}{2}$ ). Los valores obtenidos con el nuevo umbral se presentan en la [Tabla 5.19](#).

En base a los datos presentados en la [Tabla 5.19](#), se generaron las métricas de *precision*, *recall*, *accuracy* y *f1-score* presentadas en la [Tabla 5.20](#).

<i>Objetivo</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>F1-Score</i>
Similitud	0,6142	0,9352	0,6739	0,7415
Raza	0,7080	0,9055	0,7660	0,7946
Género	0,6563	0,7725	0,6840	0,7097

Tabla 5.18: Métricas de Evaluación para Comparaciones de Imágenes

<i>Objetivo</i>	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>
Similitud	1859	1526	650	317
Raza	1656	1636	364	344
Género	1572	1546	454	428

Tabla 5.19: Resumen de las matrices de confusión del análisis del umbral  $\mu + \frac{\sigma}{2}$

<i>Objetivo</i>	<i>Precision</i>	<i>Recall</i>	<i>Accuracy</i>	<i>F1-Score</i>
Similitud	0,7409	0,8543	0,7778	0,7936
Raza	0,8198	0,8280	0,8230	0,8239
Género	0,7759	0,7860	0,7795	0,7809

Tabla 5.20: Métricas de Evaluación para Comparaciones de Imágenes

Las métricas presentadas en la [Tabla 5.20](#) muestran una mejora en comparación con las presentadas en la [Tabla 5.18](#) y los resultados obtenidos para el umbral  $\mu + \frac{\sigma}{2}$  fueron satisfactorios. Se decidió utilizar  $\mu + \frac{\sigma}{2}$  como el umbral para los tres objetivos. Los valores finales se presentan en [Tabla 5.21](#)

<i>Objetivo</i>	<i>Umbral</i>
Similitud	0,32
Raza	0,88
Género	0,37

Tabla 5.21: Valores de los umbrales seleccionados

### 5.2.7. Resultados de la validación

En esta sección, se presentan los resultados del proceso de validación luego de realizada la configuración paramétrica del algoritmo evolutivo.

#### Convergencia del algoritmo

En la [Tabla 5.22](#) se presentan la distancia generacional invertida media y su desviación estandar ( $\overline{IGD} \pm std$ ), la media y desviación estándar del spacing ( $\overline{Sp} \pm std$ ), la media y desviación estándar del hipervolumen relativo ( $\overline{HvR} \pm std$ ), el valor máximo del hipervolumen relativo ( $max(HvR)$ ) y la media y desviación estándar del tiempo de ejecución ( $\overline{T}(s) \pm std$ ) para cada instancia.

$I$	$max(HvR)$	$\overline{HvR} \pm std$	$\overline{IGD} \pm std$	$\overline{Sp} \pm std$	$\overline{T}(s) \pm std$
I1	0,933	0,822 $\pm$ 0,075	0,012 $\pm$ 0,005	0,060 $\pm$ 0,029	6990 $\pm$ 501
I2	0,909	0,800 $\pm$ 0,041	0,007 $\pm$ 0,002	0,068 $\pm$ 0,026	7170 $\pm$ 533
I4	0,925	0,800 $\pm$ 0,041	0,005 $\pm$ 0,001	0,106 $\pm$ 0,075	11182 $\pm$ 1220
I6	0,841	0,695 $\pm$ 0,069	0,009 $\pm$ 0,001	0,067 $\pm$ 0,014	8048 $\pm$ 769
I7	0,968	0,812 $\pm$ 0,065	0,006 $\pm$ 0,002	0,045 $\pm$ 0,016	7795 $\pm$ 1270
I8	0,742	0,704 $\pm$ 0,024	0,006 $\pm$ 0,001	0,092 $\pm$ 0,041	7764 $\pm$ 1287

Tabla 5.22: Métricas de rendimiento por instancia

Las instancias con menor valor de hipervolumen relativo medio en la [Tabla 5.22](#) son las que tienen como objetivo las razas a las cuales el pipeline tiende a subrepresentar, como se analizó en [Subsección 5.1.3](#). En particular, las instancias con menor hipervolumen presentaban como razas objetivo medio oriente e india. Los valores de hipervolumen relativo medio son similares entre las instancias, pero se destaca la instancia I1 con un hipervolumen relativo medio de 0.822 a pesar de que cuenta con una desviación estándar más elevada. La instancia I7 obtiene resultados similares, con media y desviación estándar de hipervolumen relativo levemente inferiores a I1. Por otro lado, los valores mínimos de la media respecto al hipervolumen corresponden a las instancias I6 e I8, ambas con género objetivo femenino, mientras que I1 e I7 tienen género objetivo masculino, un resultado que es esperable teniendo en cuenta el sesgo estudiado en la [Subsección 5.1.3](#) del generador y el reconocedor hacia el género masculino. Las instancias restantes I2 e I4 mantienen valores intermedios de media y desviación estándar, resultando interesante observar que ambos se califican como razas originales de medio oriente e india, pertenecientes a la misma región geográfica. Por otro lado, las razas objetivo son blanca y negra, lo cual podría explicar los buenos resultados dado que estas razas son las más representadas.

La IGD presenta valores en su mayoría debajo del 0.01, lo cual indica una buena convergencia del algoritmo al frente de Pareto de referencia. Se destaca en este caso la instancia I4, con un valor medio y desviación estándar de IGD entre los más bajos del conjunto, manteniendo la mayor parte de los conjuntos de soluciones con un IGD en el intervalo [0,0043, 0,0064]. El spacing también presenta

valores bajos, mayormente por debajo de 0.1. Los valores de spacing cercanos a cero sugieren un buen nivel de diversidad, donde las soluciones encontradas están distribuidas de manera uniforme en el espacio latente.

En rasgos generales la instancia I4 es la más consistente en cuanto a las métricas analizadas. La peor instancia es la I8, con el segundo valor más bajo de hipervolumen relativo medio y el valor más alto de spacing, a pesar del bajo valor en IGD y desviaciones estándar en general. Los resultados para la instancia I8 pueden deberse principalmente a la edad del rostro objetivo, ya que como se analizó en la [Subsección 5.1.3](#), DeepFace presenta dificultades para clasificar rostros de adultos mayores.

Se evaluó además la evolución de la media y la desviación estándar del hipervolumen relativo a lo largo de las generaciones, como se muestra en la [Figura 5.17](#) para la instancia I2. El resto de las instancias se presentan en la [Sección C.1](#). En la [Figura 5.18](#) se muestra también la evolución de las soluciones no dominadas a través de las generaciones mediante una gráfica de tres dimensiones. Presenta las generaciones 0, 20, 50 y 99, además de la aproximación al verdadero frente de Pareto, con respectivas gráficas de corte que reflejan la evolución uno a uno de los distintos objetivos del algoritmo, demostrando que el algoritmo converge. Particularmente se presentan las [Figura 5.19](#), [Figura 5.20](#) y [Figura 5.21](#) correspondientes a la instancia I2. La evolución de las soluciones no dominadas para las instancias I1, I4, I6, I7 e I8 se presentan en la [Sección C.1](#).

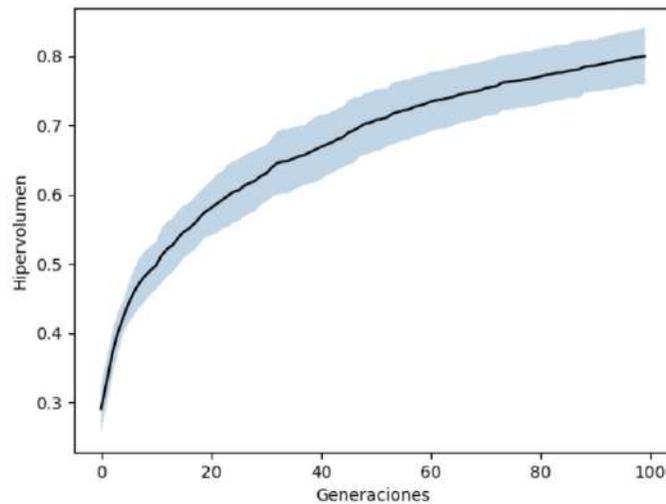


Figura 5.17: Evolución del hipervolumen para la instancia I2

Las instancias con características menos representadas por DeepFace tendieron a mostrar menores valores de hipervolumen y mayores valores de spacing. Además, el algoritmo mostró una mejor tasa de aciertos para características más representadas, reflejando el sesgo de los modelos de análisis utilizados.

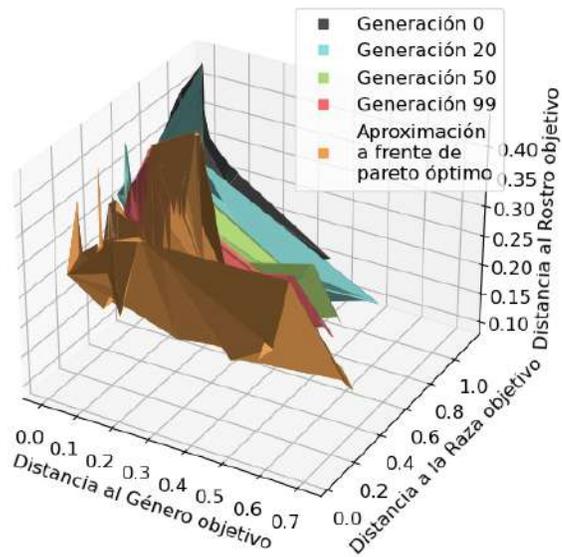


Figura 5.18: Evolución del frente de Pareto para la instancia I2

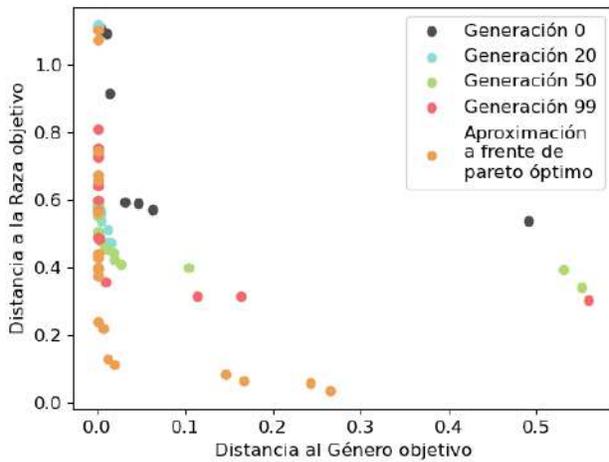


Figura 5.19: Evolución del frente de Pareto para la instancia I2 respecto a los ejes de raza y género.

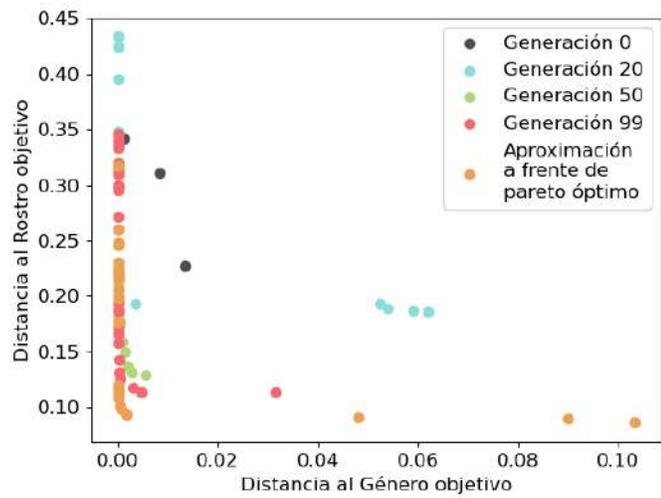


Figura 5.20: Evolución del frente de Pareto para la instancia I2 respecto a los ejes de similitud y género.

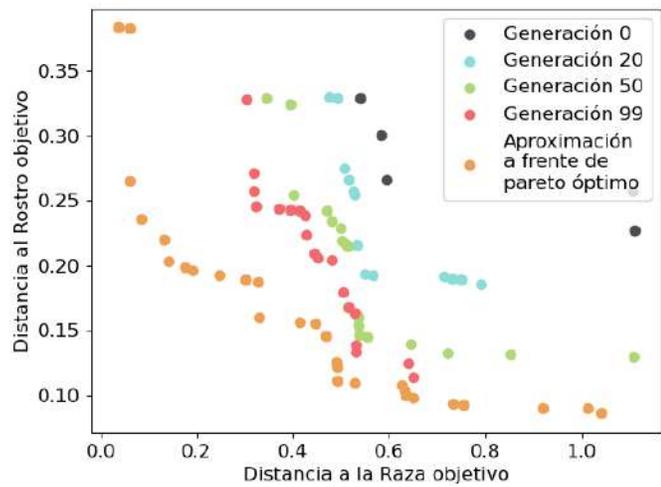


Figura 5.21: Evolución del frente de Pareto para la instancia I2 respecto a los ejes de similitud y raza.

### Análisis de la clasificación de las soluciones

Se examinó la precisión de las soluciones no dominadas obtenidas, comparando los valores de similitud, raza y género devueltos por el pipeline con los valores de umbral determinados en la [Tabla 5.21](#). Se utilizaron para el análisis todas las ejecuciones realizadas en la etapa de validación para cada instancia. Se consideraron aciertos en las características analizadas (raza, género y similitud de rostro) las imágenes que obtuvieron valores de distancia por debajo de los umbrales definidos en [Tabla 5.21](#). Se calculó la tasa de aciertos en las categorías de raza, género y similitud, determinando cuántos rostros fueron detectados como pertenecientes a la raza y género objetivo, así como cuántos fueron correctamente identificados como el individuo de la imagen objetivo.

Los resultados se presentan en la [Tabla 5.23](#), que muestra para cada instancia ( $I$ ) la tasa de aciertos para cada una de las categorías similitud, raza y género.  $A_s$  indica la tasa de rostros que fueron clasificados correctamente como el individuo de la imagen objetivo,  $A_r$  refleja la tasa de rostros que fueron clasificados correctamente como pertenecientes a la raza objetivo y  $A_g$  muestra la tasa de rostros que fueron clasificados correctamente como pertenecientes al género objetivo.

$I$	$A_s$	$A_r$	$A_g$
I1	91,27 %	71,47 %	89,09 %
I2	84,51 %	81,15 %	83,56 %
I4	76,07 %	86,48 %	87,92 %
I6	65,16 %	81,57 %	65,32 %
I7	83,23 %	72,78 %	93,48 %
I8	82,26 %	84,96 %	87,76 %

Tabla 5.23: Aciertos por instancia

La alta tasa de aciertos en todas las instancias para la mayor parte de las características refleja que el pipeline logró generar imágenes que se asemejan a las características de la imagen objetivo con características raciales y de género específicas. Las imágenes generadas logran también engañar a DeepFace.

La instancia I6 fue la que obtuvo menor cantidad de aciertos de género, tasa que se puede relacionar a que se parte de una imagen objetivo que presenta un hombre maquillado y se intenta llevar al objetivo de género femenino, dos cosas que el analizador presenta problemas para reconocer, como se vio en la [Subsección 5.1.3](#). La instancia I7 es la que presenta los mejores resultados, en parte porque su género objetivo era el masculino. Además, los buenos resultados obtenidos pueden atribuirse al hecho de que la instancia partía de una mujer con rasgos ligeramente masculinos.

La instancia I4 presenta la mayor tasa de aciertos de raza debido a que su raza objetivo era la raza blanca, que presenta un sesgo muy grande en DeepFace, como se analizó en la [Subsección 5.1.3](#). Los peores resultados provienen de las instancias I1 e I7, que tienen como raza objetivo la asiática y la latino hispana.

Los resultados sugieren que el algoritmo muestra una mejor tasa de aciertos al analizar imágenes de personas de raza blanca, particularmente hombres, lo que está en línea con el sesgo analizado en la [Subsección 5.1.3](#). Por otro lado, hay más variabilidad y errores en la identificación de personas de razas subrepresentadas y en el reconocimiento de géneros femeninos cuando la instancia también presenta características atípicas (como maquillaje en el caso de la instancia I6).

### Rostros finales generados

Se presentan las imágenes correspondientes a algunas de las soluciones finales generadas por el algoritmo evolutivo, donde se han optimizado el género y la raza objetivo según los criterios establecidos. Se presenta además para cada instancia una tabla comparativa de los valores de similitud, raza y género de cada una de las soluciones. Es importante remarcar que todas las soluciones presentadas logran engañar a DeepFace.

En la [Figura 5.22](#), se utilizó la instancia I2 como imagen objetivo, que representa a un hombre del Medio Oriente. La raza y género objetivo seleccionados fueron la raza negra y el género femenino. A la izquierda, se muestra la imagen objetivo, y a la derecha se presentan tres de las soluciones generadas por el algoritmo evolutivo.



Figura 5.22: Instancia I2 y ejemplos de soluciones generadas por el algoritmo

<i>Imagen</i>	<i>Similitud</i>	<i>Raza</i>	<i>Género</i>
1	$9,52 \times 10^{-2}$	$5,428 \times 10^{-1}$	$2,27 \times 10^{-3}$
2	$2,818 \times 10^{-1}$	$6,118 \times 10^{-1}$	$1,01 \times 10^{-4}$
3	$1,441 \times 10^{-1}$	$6,231 \times 10^{-1}$	$1,78 \times 10^{-3}$

Tabla 5.24: Valores de similitud, raza y género para I2

La [Figura 5.23](#) presenta otro ejemplo en el que se usó la instancia I6 (James Charles) como imagen objetivo y a su derecha tres de las soluciones del algoritmo evolutivo donde se usó la raza medio oriente como raza objetivo y el género femenino como género objetivo.



Figura 5.23: Instancia I6 y ejemplos de soluciones generadas por el algoritmo

<i>Imagen</i>	<i>Similitud</i>	<i>Raza</i>	<i>Género</i>
1	$1,262 \times 10^{-1}$	$7,795 \times 10^{-1}$	$8,41 \times 10^{-4}$
2	$2,635 \times 10^{-5}$	$8,795 \times 10^{-1}$	$1,03 \times 10^{-4}$
3	$3,487 \times 10^{-8}$	$9,143 \times 10^{-1}$	$1,27 \times 10^{-4}$

Tabla 5.25: Valores de similitud, raza y género para I6

La [Figura 5.24](#) presenta para la instancia I4 la imagen objetivo de una mujer india, objetivos de raza blanca y género masculino, y tres soluciones del algoritmo evolutivo.



Figura 5.24: Instancia I4 y ejemplos de soluciones generadas por el algoritmo

<i>Imagen</i>	<i>Similitud</i>	<i>Raza</i>	<i>Género</i>
1	$1,281 \times 10^{-1}$	$8,538 \times 10^{-1}$	$1,740 \times 10^{-1}$
2	$1,634 \times 10^{-1}$	$8,267 \times 10^{-1}$	$1,238 \times 10^{-1}$
3	$2,464 \times 10^{-1}$	$7,821 \times 10^{-1}$	$6,140 \times 10^{-2}$

Tabla 5.26: Valores de similitud, raza y género para I4

La [Figura 5.25](#) presenta a Marcos Mundstock (I8) como la imagen objetivo, y a su derecha tres soluciones finales generadas por el algoritmo evolutivo, donde la raza y género objetivo son raza india y género femenino.



Figura 5.25: Instancia I8 y ejemplos de soluciones generadas por el algoritmo

<i>Imagen</i>	<i>Similitud</i>	<i>Raza</i>	<i>Género</i>
1	$2,279 \times 10^{-1}$	$4,614 \times 10^{-1}$	$4,000 \times 10^{-4}$
2	$2,715 \times 10^{-8}$	$7,179 \times 10^{-1}$	$1,000 \times 10^{-4}$
3	$1,862 \times 10^{-1}$	$7,222 \times 10^{-1}$	$3,700 \times 10^{-3}$

Tabla 5.27: Valores de similitud, raza y género para I8

Para la instancia que utiliza a Whoopi Goldberg (I7) como rostro objetivo, se utilizó una raza objetivo latina hispana y un género objetivo masculino. La [Figura 5.26](#) presenta el rostro objetivo junto con tres soluciones generadas por el algoritmo evolutivo.

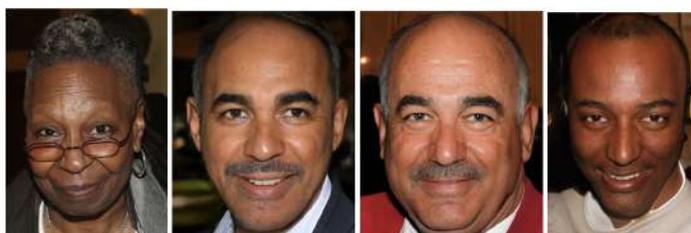


Figura 5.26: Instancia I7 y ejemplos de soluciones generadas por el algoritmo

<i>Imagen</i>	<i>Similitud</i>	<i>Raza</i>	<i>Género</i>
1	$1,961 \times 10^{-1}$	$8,156 \times 10^{-1}$	$1,667 \times 10^{-2}$
2	$2,116 \times 10^{-1}$	$7,641 \times 10^{-1}$	$1,830 \times 10^{-2}$
3	$1,209 \times 10^{-1}$	$9,875 \times 10^{-1}$	$6,920 \times 10^{-2}$

Tabla 5.28: Valores de similitud, raza y género para I7

Por último, la [Figura 5.27](#) presenta a Margot Robbie (I1) y a su derecha tres soluciones finales generadas por el algoritmo evolutivo, donde la raza y género objetivo son raza asiática y género masculino.



Figura 5.27: Instancia I1 y ejemplos de soluciones generadas por el algoritmo

<i>Imagen</i>	<i>Similitud</i>	<i>Raza</i>	<i>Género</i>
1	$1,334 \times 10^{-1}$	$5,943 \times 10^{-1}$	$5,470 \times 10^{-2}$
2	$1,559 \times 10^{-1}$	$6,472 \times 10^{-1}$	$1,660 \times 10^{-2}$
3	$1,068 \times 10^{-1}$	$7,282 \times 10^{-1}$	$1,551 \times 10^{-1}$

Tabla 5.29: Valores de similitud, raza y género para la instancia I1

A partir de las imágenes presentadas en las figuras [5.22](#), [5.23](#), [5.24](#), [5.25](#), [5.26](#) y [5.27](#) se corrobora que el algoritmo es capaz de producir imágenes que resultan cercanas a los objetivos planteados al ser analizadas por el ojo humano. Se destaca que las imágenes logran engañar a DeepFace al mantenerse los valores de similitud, raza y género por debajo de los umbrales establecidos en la [Tabla 5.21](#).



## Capítulo 6

# Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones del proyecto y los posibles trabajos futuros.

### 6.1. Conclusiones

El proyecto se centró en el estudio, diseño, implementación y validación de estrategias evolutivas para la exploración del espacio latente de las GANs. El objetivo del proyecto fue generar muestras sintéticas que cuenten con un conjunto de características o atributos deseados y sean a la vez similares a una imagen objetivo. El trabajo presentado partió de la implementación de un proyecto existente que buscó generar imágenes similares a una imagen objetivo, garantizando también una raza del rostro objetivo. El enfoque añadió un nuevo objetivo al proceso: la inclusión del género como un atributo a ser considerado. De esta manera, no solo se buscó que los rostros generados compartieran características faciales con la imagen objetivo y cumplieran con la raza especificada, sino que también se asegurara que el género del rostro generado coincidiera con el deseado.

Para el desarrollo del proyecto fue fundamental recurrir a la literatura relacionada para adquirir una comprensión del campo. La revisión incluyó la exploración del estado del arte sobre la exploración del espacio latente de GANs y el uso de algoritmos evolutivos para la generación de rostros artificiales. A través de la revisión, se identificó que muchos estudios utilizaban StyleGAN2. Se optó por utilizar StyleGAN3 debido a su menor presencia en la literatura y su mejora frente a StyleGAN2. El algoritmo evolutivo NSGA-II, descrito en diversos estudios, se analizó como una opción y su aplicación en el contexto estudiado mostró resultados positivos. La revisión bibliográfica no sólo proporcionó una sólida base teórica, sino que también orientó las decisiones metodológicas del proyecto.

El pipeline del proyecto incluye tres componentes principales: StyleGAN3 para la generación de imágenes, DeepFace para el análisis de rostros y DEAP para la implementación del algoritmo evolutivo. El proceso comienza con la generación de una imagen por StyleGAN3, que luego es procesada y analizada por RetinaFace para la detección y recorte de rostros. Si un rostro es identificado, se extraen sus embeddings y se calculan los vectores de raza y género usando DeepFace. Las medidas obtenidas a partir de la distancia entre los vectores generados y los objetivos para la raza, el género y la similitud se utilizan para evaluar el fitness del individuo. El pipeline emplea operadores de cruce y mutación para evolucionar la población de soluciones a lo largo de 100 generaciones, optimizando múltiples objetivos simultáneamente. Para la ejecución del algoritmo se reemplaza la función de activación softmax en DeepFace por una función sigmoide, mejorando así la distribución de los valores de probabilidad de género. La modificación permite una dispersión más uniforme de los valores, logrando una clasificación mas acorde con el género real de los rostros analizados.

Luego de tener una mayor comprensión del campo se realizó un análisis de sesgo del pipeline generado por StyleGAN3 y Facebook DeepFace. Se generaron imágenes utilizando vectores latentes aleatorios en StyleGAN3 y se empleó DeepFace para analizar los rostros generados. Se validó OpenCV para la detección de rostros, que no reconoció cerca del 10 % de las imágenes generadas. Las imágenes no reconocidas eran rostros claros a simple vista, lo que llevó a analizar otros detectores de rostros, entre ellos SSD, MTCNN, RetinaFace y Yunet. RetinaFace demostró ser el más efectivo, mejorando significativamente la detección en comparación con OpenCV, con un tiempo de ejecución aceptable. Se observó que el pipeline conformado por StyleGAN3 y DeepFace tiene un sesgo marcado hacia la generación de rostros de hombres adultos de raza blanca. Del total de imágenes reconocidas, el 57 % fueron clasificadas como personas de raza blanca, el 82.9 % como adultos y el 63 % como masculinos.

Para determinar el origen del sesgo, se analizaron las imágenes del conjunto de datos de entrenamiento de StyleGAN3 (Flickr-Faces-HQ) y los resultados de DeepFace utilizando la base de datos etiquetada UTKFace. Se encontró que ambos componentes contribuyen al sesgo: StyleGAN3 por la distribución de su conjunto de entrenamiento y DeepFace por su tendencia a sobrerrepresentar ciertas categorías, especialmente en género y edad.

Para la experimentación, se seleccionaron nueve imágenes de rostros teniendo en cuenta la diversidad en género, raza y edad. Las imágenes incluyen géneros masculinos y femeninos de diversas edades y razas, excluyendo a niños para evitar dificultades en la determinación del género. Se evaluaron diferentes porcentajes de expansión alrededor del rostro para mejorar la precisión en la identificación de género y raza. El análisis demostró que una expansión del 40 % proporcionó los mejores resultados a la hora de ser analizada por DeepFace.

Además, se ajustaron los parámetros del algoritmo evolutivo correspondientes al tamaño de la población y las probabilidades de mutación y cruzamiento, para optimizar la generación de rostros. Se experimentó con diferentes configuraciones de los parámetros para encontrar la combinación que permitió al algoritmo generar rostros con alta similitud a las imágenes objetivo y que pre-

sentaron características de la raza y género objetivo. Se analizaron diversas configuraciones de rostro, género y raza para evaluar cómo diferentes ajustes de los parámetros afectan la precisión en la creación de rostros con las características deseadas. Durante el análisis de la configuración paramétrica del algoritmo evolutivo, se evaluaron 6 instancias con 8 configuraciones distintas cada una, ejecutadas 20 veces por configuración. Se usó el test de Kolmogorov-Smirnov para verificar la normalidad de la distribución de los resultados, confirmando que los datos seguían una distribución normal. Se aplicó el test de rangos de Friedman para comparar el rango promedio de cada configuración. Los resultados se organizaron para identificar la mejor configuración.

Para establecer umbrales adecuados para determinar la similitud entre el rostro generado y los objetivos de raza, género y similitud facial, se llevaron a cabo comparaciones utilizando las bases de datos Labeled Faces in the Wild para la similitud facial y FairFace para la raza y el género. Inicialmente, se calcularon umbrales utilizando la media más una desviación estándar. Sin embargo, debido a un alto número de falsos positivos, se optó por utilizar la media más la mitad de la desviación estándar como el umbral definitivo. El umbral elegido mejoró significativamente las métricas de precisión, validando así la efectividad de los nuevos umbrales.

En la fase de validación, después de configurar los hiperparámetros y ejecutar el algoritmo en 60 ejecuciones independientes para cada instancia, se construyó una aproximación al verdadero frente de Pareto. Se evaluaron los resultados con medidas del hipervolumen relativo, la distancia generacional invertida y el spacing, que indicaron una buena convergencia del algoritmo hacia la aproximación al verdadero frente de Pareto. Se observó que las instancias con características menos representadas por DeepFace tendieron a mostrar menores valores de hipervolumen y mayores valores de spacing. Además, el algoritmo mostró una mejor tasa de aciertos para características más representadas, reflejando el sesgo de los modelos de análisis utilizados.

Dado que el algoritmo debió cumplir tres objetivos diferentes, se permitió una exploración guiada del espacio latente de la GAN, resultando en la creación de imágenes visualmente similares a la imagen objetivo, con los atributos de raza y género correspondientes, y reconocidas como la misma persona por DeepFace.

## 6.2. Trabajo futuro

En futuras investigaciones, se plantea agregar un objetivo adicional relacionado con el rango etario. El objetivo adicional planteado generaría que el algoritmo evolutivo no solo optimice las imágenes generadas en términos de similitud, raza y género, sino también en relación con la edad de las personas. Debido al pobre desempeño de DeepFace en términos de clasificación de edad, en especial para los rangos etarios extremadamente altos o bajos, sería útil considerar otro modelo con mejores capacidades para el análisis de edad.

Por otro lado, se plantea la utilización de otro algoritmo evolutivo en lugar de NSGA-II y comparar la calidad de resultados obtenidos por ambos algoritmos y los tiempos de ejecución. Uno de los algoritmos que se considera pertinente para investigaciones futuras es el llamado Multi-Objective Covariance Matrix Adaptation Evolution Strategy, ya que su variación mono-objetivo fue utilizada para trabajos relacionados, como se mencionó en el [Capítulo 3](#), brindando buenos resultados.

Además, el sesgo analizado en la [Subsección 5.1.3](#) pudo haber influenciado en gran medida la calidad de imágenes generadas. Por esta razón, se plantea como trabajo futuro evaluar la utilización de un generador o analizador con menor sesgo, o incluso considerar el entrenamiento de una red como StyleGAN3 con una base de datos donde la distribución de rostros en cuanto a raza y género sea equitativa.

Por otro lado, se plantea modificar el espacio latente explorado o la población inicial utilizada con el fin de buscar una exploración más eficaz y eficiente. La utilización del espacio latente inicial  $Z$  de StyleGAN3 probó ser adecuada para generar imágenes que cumplieran los objetivos planteados. Sin embargo, la utilización de los espacios intermedios  $W$  o  $W+$  como espacio latente a explorar en futuras investigaciones puede ser beneficioso para encontrar mejores soluciones, como explicaron [Karras y cols. \(2018\)](#).

Por último, se propone utilizar métodos de inicialización inteligente de individuos para la población inicial, para mejorar la calidad de las soluciones. También se plantea como trabajo futuro tomar un muestreo de imágenes de diferentes razas y géneros y realizar un pre análisis con técnicas como descenso por gradiente para la inicialización de la población con imágenes más cercanas a los objetivos planteados.

# Referencias

- Correa, J., Mignaco, J., Rey, G., Machín, B., Nesmachnow, S., y Toutouh, J. (2023). Multiobjective evolutionary search of the latent space of generative adversarial networks for human face generation. En *Proceedings of the companion conference on genetic and evolutionary computation* (p. 1768–1776). Association for Computing Machinery.
- Deb, K. (2001). *Multiobjective optimization using evolutionary algorithms*. Wiley. (Libro)
- Deb, K. (2004). Genetic algorithms for optimization. En D. Kundu y A. Basu (Eds.), *Statistical computing: Existing methods and recent developments* (pp. 85–123). Narosa Publishing House.
- Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.
- Developers, D. (2024). deap.tools.emo — deap documentation [Manual de software informático]. Descargado de [https://deap.readthedocs.io/en/master/\\_modules/deap/tools/emo.html#selTournamentDCD](https://deap.readthedocs.io/en/master/_modules/deap/tools/emo.html#selTournamentDCD) (Fecha de acceso: 2024-09-05)
- Developers, U. (2024). Uniform manifold approximation and projection [Manual de software informático]. <https://umap-learn.readthedocs.io/en/latest/performance.html>. (Fecha de acceso: 2024-07-11)
- Eiben, A., y Smith, J. (2015). *Introduction to evolutionary computing*. Springer-Verlag.
- Eshelman, L. J., y Schaffer, J. D. (1993). Real-coded genetic algorithms and interval-schemata. En L. DARRELL (Ed.), *Foundations of genetic algorithms* (Vol. 2, p. 187-202). Elsevier.
- Fernandes, P., Correia, J., y Machado, P. (2020). *Evolutionary latent space exploration of generative adversarial networks* (Inf. Téc.). Portugal: Centro de Informatica y Sistemas de la Universidad de Coimbra, Departamento de Engenharia Informatica, Universidad de Coimbra.

- Forrest, S. (1996). Genetic algorithms. *ACM Computing Surveys*, 28(1), 77–80.
- Galatolo, F., Cimino, M., y Vaglini, G. (2021). *Generating images from caption and viceversa via clip-guided generative latent space search* (Inf. Téc.). Italia: Departamento de Ingeniería Informática, Universidad de Pisa.
- Holland, J. (1975). *Adaptation in natural and artificial systems*. Universidad de Michigan Press.
- Huang, G., Mattar, M., Berg, T., y Learned-Miller, E. (2007). *Labeled faces in the wild: A database for studying face recognition in unconstrained environments* (Inf. Téc.). Universidad de Massachusetts, Amherst. Descargado de <http://vis-www.cs.umass.edu/lfw/> (Reporte técnico 07-49, fecha de acceso: 2024-03-14)
- Ilachinsky, A. (1997). *Irreducible semi-autonomous adaptive combat (isaac): An artificial-life approach to land combat* (Research Memorandum n.º CRM 97-61.10). Estados Unidos: Center for Naval Analyses.
- Joglekar, A., y Tungare, M. (2001). *Genetic algorithms and their use in the design of evolvable hardware* (Inf. Téc.). India: Fr. Conceicao Rodrigues College of Engineering.
- Karkkainen, K., Joo, K., y Matusik, W. (2021). *Fairface: Face attribute dataset for assessing bias*. <https://github.com/joojs/fairface>. Descargado de <https://github.com/joojs/fairface> (Repositorio de GitHub, fecha de acceso: 2024-03-14)
- Karras, T., Aila, T., Laine, S., y Lehtinen, J. (2019). *Flickr-faces-hq dataset*. <https://github.com/NVlabs/ffhq-dataset>. Descargado de <https://github.com/NVlabs/ffhq-dataset> (Repositorio de GitHub, Fecha de acceso: 2024-03-14)
- Karras, T., Laine, S., y Aila, T. (2018). A style-based generator architecture for generative adversarial networks. En *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (cvpr)*.
- Koza, J., y Poli, R. (1997). Genetic programming. En A. Kent y J. G. Williams (Eds.), *Encyclopedia of science and technology* (Vol. 39, pp. 29–43).
- learn Developers, S. (2024). *t-distributed stochastic neighbor embedding*. <https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html>. (Fecha de acceso: 2024-05-04)
- Machín, B., Nesmachnow, S., y Toutouh, J. (2021). Evolutionary latent space search for driving human portrait generation. *2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 1-6.

- Machín, B., Nesmachnow, S., y Toutouh, J. (2022). Multi-target evolutionary latent space search of a generative adversarial network for human face generation. En *Proceedings of the genetic and evolutionary computation conference companion* (p. 1878–1886). Association for Computing Machinery.
- Metz, L., Poole, B., Pfau, D., y Sohl-Dickstein, J. (2017). Unrolled generative adversarial networks. En *Proceedings of the international conference on learning representations (iclr)*.
- Mitchell, M. (1996). *An introduction to genetic algorithms*. The MIT Press.
- Mohsienuddin, S., y Sabri, M. (2020). Facial recognition technology. *SSRN Electronic Journal*, 7(6), 176-184.
- Morrison, J. (1998). *Co-evolution and genetic algorithms* (Tesis de Master). Escuela de Ciencias de la Computación, Universidad de Carleton.
- Neumann, A., Pyromallis, C., y Alexander, B. (2018). *Evolution of images with diversity and constraints using a generative adversarial network* (Inf. Téc.). Australia: Optimización y Logística, Escuela de Ciencias de la Computación, Universidad de Adelaida.
- Pitiddu, O., y Veiga, R. (2022). Diseño de clasificadores de dígitos manuscritos generados por redes gan. En *2022 ieee biennial congress of argentina (argencon)* (p. 1-7).
- Rothe, R., Timofte, R., y Van Gool, L. (2016). *Imdb-wiki – 500k+ face images with age and gender labels*. <https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/>. Descargado de <https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/> (Repositorio de datos, fecha de acceso: 2024-03-14)
- Roziere, B., Teytaud, F., Hosu, V., Lin, H., Rapin, J., Zameshina, M., y Teytaud, O. (2020). Evolgan: Evolutionary generative adversarial networks. En *Proceedings of the asian conference on computer vision (accv)*.
- Serengil, S., y Ozpinar, A. (2020). *Deepface: A lightweight face recognition and facial attribute analysis framework for python*. <https://github.com/serengil/deepface>. Descargado de <https://github.com/serengil/deepface> (Repositorio de GitHub, fecha de acceso: 2023-12-12)
- Taijman, Y., Yang, M., Ranzato, M., y Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. En *Proceedings of the ieee conference on computer vision and pattern recognition (cvpr)*.
- Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., y Wang, F. (2017). Generative adversarial networks: Introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*.
- Whitley, D. (2001). An overview of evolutionary algorithms: Practical issues and common pitfalls. *Journal of Information and Software Technology*, 43(14), 817–831.

- Yu, X., y Gen, M. (2010). *Introduction to evolutionary algorithms*. Springer.
- Zhang, Z., Song, Y., y Qi, H. (2017). *Utkface: A large-scale face dataset for age, gender, and ethnicity*. <https://susanqq.github.io/UTKFace>. (Fecha de acceso: 2024-09-04)
- Zitzler, E., Deb, K., y Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2), 173-195.

## Anexo A

# Tablas de configuración paramétrica

### A.1. Configuración paramétrica del algoritmo

#### A.1.1. Test de rangos Friedman

	<i>Configuración</i>							
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
$\sum_{Rango}$	83	91	102	113	62	87	77	105
$P_{Rango}$	4.15	4.55	5.10	5.65	3.10	4.35	3.85	5.25
<i>Rango</i>	3.85	3.45	2.90	<b>2.35</b>	<b>4.90</b>	3.65	4.15	2.75

Tabla A.1: Test de Friedman para I5 con objetivo de raza medio oriente y género masculino

	<i>Configuración</i>							
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
$\sum_{Rango}$	81	107	93	110	71	88	68	102
$P_{Rango}$	4.05	5.35	4.65	5.50	3.55	4.40	3.40	5.10
<i>Rango</i>	3.95	2.65	3.35	<b>2.50</b>	4.45	3.60	<b>4.60</b>	2.90

Tabla A.2: Test de Friedman para I3 con objetivos de raza india y género femenino

	<i>Configuración</i>							
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
$\sum_{Rango}$	95	109	105	111	65	92	59	84
$P_{Rango}$	4.75	5.45	5.25	5.55	3.25	4.60	2.95	4.20
<i>Rango</i>	3.25	2.55	2.75	<b>2.45</b>	4.75	3.40	<b>5.05</b>	3.80

Tabla A.3: Test de Friedman para I3 con objetivo de raza asiatica y género femenino

	<i>Configuración</i>							
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
$\sum_{Rango}$	106	87	84	112	60	102	63	106
$P_{Rango}$	5.30	4.35	4.20	5.60	3.00	5.10	3.15	5.30
<i>Rango</i>	2.70	3.65	3.80	<b>2.40</b>	<b>5.00</b>	2.90	4.85	2.70

Tabla A.4: Test de Friedman para I9 con objetivos de raza latino-hispana y género masculino

	<i>Configuración</i>							
	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
$\sum_{Rango}$	80	115	87	128	65	94	72	79
$P_{Rango}$	4.00	5.75	4.35	6.40	3.25	4.70	3.60	3.95
<i>Rango</i>	4.00	2.25	3.65	<b>1.60</b>	<b>4.75</b>	3.30	4.40	4.05

Tabla A.5: Test de Friedman para I9 con objetivos de raza blanca y género masculino

### A.1.2. Resultados de configuración paramétrica

<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>Configuración</i>					<i>C8</i>
			<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>	
0.720	0.731	0.703	0.720	0.704	0.746	0.675	0.720	
0.780	0.686	0.787	0.793	0.644	0.771	0.731	0.707	
0.725	0.688	0.723	0.819	0.692	0.763	0.685	0.749	
0.761	0.735	0.742	0.785	0.747	0.644	0.726	0.726	
0.742	0.768	0.742	0.810	0.697	0.713	0.712	0.776	
0.666	0.766	0.751	0.780	0.684	0.731	0.676	0.789	
0.703	0.713	0.776	0.769	0.744	0.714	0.693	0.704	
0.702	0.745	0.790	0.776	0.740	0.682	0.670	0.802	
0.723	0.772	0.678	0.864	0.717	0.774	0.738	0.766	
0.709	0.731	0.679	0.850	0.700	0.717	0.681	0.713	
0.872	0.783	0.742	0.785	0.681	0.691	0.640	0.716	
0.663	0.733	0.720	0.777	0.725	0.691	0.759	0.706	
0.829	0.719	0.735	0.782	0.716	0.788	0.700	0.724	
0.730	0.746	0.679	0.821	0.709	0.718	0.714	0.730	
0.778	0.696	0.688	0.800	0.676	0.693	0.706	0.730	
0.662	0.709	0.732	0.758	0.675	0.834	0.703	0.782	
0.813	0.751	0.775	0.748	0.732	0.719	0.780	0.668	
0.745	0.769	0.773	0.829	0.733	0.782	0.716	0.749	
0.690	0.761	0.808	0.681	0.636	0.768	0.699	0.728	
0.722	0.750	0.758	0.755	0.721	0.668	0.664	0.693	

Tabla A.6: Resultados de configuración paramétrica para la instancia I5 con objetivos de raza negra y género masculino

<i>Configuración</i>							
<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
0.558	0.535	0.627	0.734	0.669	0.547	0.691	0.634
0.532	0.539	0.530	0.673	0.510	0.714	0.632	0.596
0.708	0.616	0.811	0.820	0.532	0.641	0.779	0.609
0.682	0.636	0.652	0.645	0.660	0.680	0.747	0.748
0.637	0.548	0.624	0.573	0.586	0.518	0.521	0.765
0.517	0.611	0.664	0.747	0.569	0.600	0.643	0.743
0.521	0.738	0.648	0.659	0.608	0.679	0.499	0.627
0.567	0.609	0.677	0.642	0.643	0.670	0.680	0.634
0.619	0.671	0.666	0.587	0.530	0.702	0.613	0.726
0.658	0.663	0.843	0.799	0.577	0.560	0.614	0.694
0.645	0.610	0.636	0.795	0.519	0.750	0.570	0.718
0.680	0.679	0.668	0.804	0.822	0.500	0.659	0.661
0.547	0.643	0.520	0.754	0.540	0.667	0.587	0.708
0.676	0.660	0.682	0.736	0.619	0.660	0.629	0.648
0.527	0.536	0.703	0.581	0.535	0.720	0.587	0.776
0.654	0.760	0.710	0.702	0.673	0.653	0.702	0.631
0.596	0.715	0.755	0.710	0.661	0.537	0.618	0.780
0.735	0.707	0.653	0.646	0.569	0.534	0.605	0.564
0.632	0.681	0.561	0.621	0.551	0.708	0.487	0.856
0.755	0.752	0.613	0.598	0.635	0.730	0.590	0.605

Tabla A.7: Resultados de configuración paramétrica para la instancia I5 con objetivos de raza de medio oriente y género masculino

<i>Configuración</i>							
<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
0.692	0.739	0.678	0.788	0.703	0.717	0.718	0.707
0.571	0.715	0.728	0.790	0.705	0.873	0.689	0.667
0.606	0.719	0.615	0.834	0.742	0.712	0.643	0.585
0.740	0.657	0.744	0.683	0.606	0.679	0.675	0.659
0.661	0.671	0.708	0.703	0.665	0.636	0.666	0.676
0.732	0.770	0.703	0.771	0.639	0.554	0.555	0.693
0.714	0.806	0.648	0.656	0.675	0.682	0.732	0.719
0.694	0.675	0.733	0.633	0.648	0.784	0.716	0.777
0.704	0.647	0.686	0.939	0.605	0.712	0.702	0.705
0.667	0.726	0.746	0.666	0.657	0.643	0.686	0.709
0.700	0.759	0.638	0.624	0.644	0.708	0.644	0.725
0.610	0.699	0.761	0.790	0.697	0.641	0.615	0.644
0.680	0.633	0.659	0.652	0.689	0.628	0.601	0.733
0.704	0.750	0.705	0.675	0.655	0.654	0.664	0.848
0.672	0.691	0.623	0.832	0.701	0.687	0.709	0.684
0.764	0.735	0.858	0.780	0.638	0.659	0.538	0.775
0.746	0.824	0.827	0.721	0.688	0.771	0.572	0.641
0.677	0.763	0.699	0.822	0.705	0.798	0.594	0.697
0.663	0.644	0.625	0.666	0.637	0.676	0.614	0.679
0.709	0.685	0.599	0.621	0.686	0.662	0.641	0.749

Tabla A.8: Resultados de configuración paramétrica para la instancia I3 con objetivos de raza india y género femenino

<i>Configuración</i>							
<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
0.691	0.743	0.597	0.799	0.622	0.678	0.569	0.596
0.643	0.689	0.616	0.815	0.612	0.622	0.663	0.598
0.608	0.610	0.591	0.724	0.649	0.630	0.583	0.729
0.601	0.740	0.660	0.633	0.574	0.643	0.609	0.632
0.610	0.625	0.662	0.701	0.515	0.603	0.592	0.669
0.635	0.729	0.753	0.711	0.629	0.680	0.625	0.603
0.767	0.623	0.713	0.754	0.601	0.787	0.611	0.635
0.612	0.710	0.622	0.613	0.561	0.743	0.655	0.689
0.616	0.727	0.624	0.676	0.651	0.578	0.721	0.717
0.637	0.679	0.580	0.725	0.632	0.630	0.646	0.628
0.607	0.670	0.620	0.660	0.618	0.622	0.574	0.713
0.635	0.656	0.613	0.739	0.616	0.753	0.634	0.672
0.650	0.627	0.551	0.698	0.570	0.612	0.634	0.571
0.652	0.724	0.675	0.638	0.575	0.644	0.718	0.618
0.592	0.669	0.722	0.672	0.715	0.700	0.546	0.576
0.572	0.660	0.706	0.667	0.636	0.657	0.554	0.635
0.710	0.622	0.660	0.726	0.613	0.551	0.579	0.638
0.612	0.690	0.658	0.661	0.660	0.752	0.755	0.631
0.752	0.636	0.582	0.698	0.638	0.602	0.631	0.600
0.563	0.589	0.732	0.906	0.677	0.602	0.611	0.643

Tabla A.9: Resultados de configuración paramétrica para la instancia I9 con género objetivo masculino y raza objetivo blanca

<i>Configuración</i>							
<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
0.709	0.737	0.672	0.676	0.790	0.824	0.699	0.700
0.769	0.670	0.708	0.782	0.749	0.781	0.741	0.787
0.675	0.711	0.791	0.726	0.699	0.701	0.822	0.728
0.788	0.629	0.702	0.743	0.672	0.738	0.733	0.698
0.777	0.781	0.731	0.718	0.677	0.734	0.700	0.732
0.698	0.760	0.732	0.889	0.716	0.723	0.586	0.695
0.729	0.716	0.811	0.788	0.701	0.776	0.663	0.756
0.694	0.666	0.737	0.749	0.847	0.755	0.846	0.606
0.832	0.795	0.743	0.699	0.726	0.893	0.780	0.828
0.845	0.902	0.844	0.746	0.736	0.638	0.601	0.695
0.732	0.705	0.696	0.749	0.599	0.733	0.683	0.755
0.704	0.721	0.732	0.795	0.695	0.730	0.762	0.899
0.813	0.711	0.748	0.743	0.806	0.798	0.712	0.806
0.810	0.747	0.734	0.779	0.710	0.764	0.693	0.740
0.739	0.819	0.756	0.764	0.734	0.796	0.693	0.776
0.721	0.756	0.635	0.851	0.711	0.665	0.668	0.733
0.809	0.669	0.677	0.805	0.730	0.721	0.621	0.817
0.759	0.680	0.757	0.880	0.659	0.676	0.716	0.786
0.803	0.707	0.664	0.817	0.646	0.692	0.713	0.660
0.744	0.743	0.794	0.734	0.588	0.752	0.663	0.764

Tabla A.10: Resultados de configuración paramétrica para la instancia I9 con género objetivo masculino y raza objetivo latino-hispana

<i>Configuración</i>							
<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>	<i>C6</i>	<i>C7</i>	<i>C8</i>
0.716	0.767	0.631	0.839	0.698	0.696	0.682	0.654
0.737	0.842	0.785	0.675	0.697	0.686	0.707	0.785
0.623	0.671	0.837	0.787	0.676	0.901	0.699	0.794
0.750	0.685	0.823	0.847	0.687	0.769	0.646	0.605
0.684	0.759	0.829	0.824	0.767	0.837	0.614	0.802
0.857	0.874	0.588	0.810	0.663	0.649	0.760	0.614
0.727	0.779	0.874	0.940	0.646	0.733	0.641	0.651
0.840	0.899	0.801	0.665	0.674	0.717	0.698	0.845
0.877	0.687	0.771	0.771	0.689	0.649	0.674	0.628
0.763	0.763	0.828	0.753	0.677	0.717	0.697	0.685
0.954	0.836	0.718	0.720	0.614	0.850	0.684	0.790
0.697	0.788	0.854	0.712	0.772	0.713	0.702	0.868
0.702	0.802	0.744	0.866	0.741	0.714	0.620	0.656
0.768	0.621	0.825	0.961	0.623	0.677	0.722	0.652
0.615	0.797	0.748	0.701	0.618	0.740	0.781	0.855
0.702	0.919	0.655	0.758	0.649	0.740	0.624	0.734
0.868	0.906	0.872	0.729	0.712	0.710	0.633	0.644
0.719	0.712	0.578	0.795	0.674	0.880	0.593	0.855
0.722	0.749	0.910	0.814	0.685	0.803	0.918	0.770
0.870	0.760	0.695	0.784	0.855	0.657	0.686	0.777

Tabla A.11: Resultados de configuración paramétrica para la instancia I3 con género objetivo femenino y raza objetivo asiática

## Anexo B

# Recorte de imágenes

### B.1. Análisis de expansión de imágenes

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Blanco	4.56	5.05	4.83	4.52	4.50
Asiático	-20.56	-28.54	-34.76	-38.93	-34.46
Indio	-22.85	-29.85	-35.68	-39.62	-35.91
Negro	-29.86	-38.50	-45.24	-50.41	-45.81
Medio Oriental	-9.31	-13.60	-17.72	-21.34	-18.92
Latino/Hispano	-8.76	-14.11	-19.60	-23.62	-21.32

Tabla B.1: Resultados de distribución de raza según su porcentaje de expansión para la instancia I5

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Blanco	1.35	2.31	1.88	1.86	1.69
Asiático	-2.15	-5.93	-2.60	-3.52	-2.60
Indio	-5.53	-7.91	-5.99	-6.41	-5.39
Negro	-6.68	-10.97	-8.63	-9.35	-7.95
Medio Oriental	-2.07	-2.33	-2.05	-2.33	-2.17
Latino/Hispano	-0.94	-1.71	-1.23	-1.16	-0.79

Tabla B.2: Resultados de distribución de raza según su porcentaje de expansión para la instancia I6

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Blanco	1.09	1.90	1.91	1.63	1.79
Asiático	-2.93	-6.37	-6.61	-5.62	-5.29
Indio	-3.99	-7.42	-6.54	-5.62	-5.42
Negro	-5.96	-11.19	-10.16	-8.95	-8.86
Medio Oriental	-1.30	-2.80	-2.19	-1.86	-1.63
Latino/Hispano	-1.05	-3.29	-2.89	-2.50	-2.33

Tabla B.3: Resultados de distribución de raza según su porcentaje de expansión para la instancia I8

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Asiático	0.84	0.45	0.13	-0.23	-0.18
Indio	-4.54	-8.47	-13.36	-16.01	-16.16
Negro	-10.68	-17.06	-26.15	-31.12	-29.86
Blanco	-8.19	-11.59	-16.89	-20.25	-20.35
Medio Oriental	-12.54	-16.88	-23.90	-29.11	-27.57
Latino/Hispano	-5.18	-8.62	-14.00	-16.40	-16.10

Tabla B.4: Resultados de distribución de raza según su porcentaje de expansión para la instancia I9

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Blanco	1.58	2.68	2.76	3.18	3.65
Asiático	-6.25	-10.64	-11.71	-13.91	-14.64
Indio	-7.73	-13.23	-14.33	-17.29	-17.03
Negro	-10.61	-17.74	-19.48	-22.72	-22.88
Medio Oriental	-2.98	-5.68	-6.36	-7.80	-7.08
Latino/Hispano	-2.36	-4.60	-4.99	-6.56	-6.30

Tabla B.5: Resultados de distribución de raza según su porcentaje de expansión para la instancia I1

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Negro	5.68	6.48	5.62	5.29	5.34
Asiático	-5.15	-6.77	-5.31	-5.22	-5.87
Indio	-3.39	-3.24	-1.57	-0.98	-1.77
Blanco	-10.94	-13.52	-12.10	-11.42	-12.36
Medio Oriental	-12.29	-14.99	-13.03	-12.19	-12.84
Latino/Hispano	-4.33	-6.46	-5.63	-5.57	-6.28

Tabla B.6: Resultados de distribución de raza según su porcentaje de expansión para la instancia I7

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Negro	9.81	6.03	6.15	7.18	6.17
Asiático	-15.15	-16.51	-15.19	-16.94	-16.23
Indio	-10.46	-13.25	-10.94	-11.37	-11.08
Blanco	-23.17	-23.05	-21.00	-22.52	-20.63
Medio Oriental	-23.57	-20.30	-18.31	-19.61	-17.08
Latino/Hispano	-13.33	-15.12	-13.84	-14.72	-13.57

Tabla B.7: Resultados de distribución de raza según su porcentaje de expansión para la instancia I3

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Latino/Hispano	0.16	-0.35	-0.41	-0.21	-0.51
Asiático	-0.82	-0.09	-0.12	-0.27	-0.09
Indio	-0.19	0.09	0.12	0.14	-0.09
Negro	-0.51	-0.08	-0.45	-0.90	-0.45
Blanco	-0.73	-1.08	-1.04	-0.69	-1.01
Medio Oriental	-0.79	-1.06	-0.77	-0.30	-0.45

Tabla B.8: Resultados de distribución de raza según su porcentaje de expansión para la instancia I2

<i>Raza</i>	<i>E(0 %)</i>	<i>E(20 %)</i>	<i>E(30 %)</i>	<i>E(40 %)</i>	<i>E(50 %)</i>
Indio	1.44	1.49	1.69	1.41	1.32
Asiático	-2.72	-2.79	-3.37	-3.26	-2.74
Negro	-1.61	-2.58	-2.59	-3.19	-2.97
Blanco	-3.58	-3.23	-3.67	-3.16	-2.69
Medio Oriental	-3.89	-2.16	-2.41	-1.30	-1.00
Latino/Hispano	-0.41	-1.33	-1.43	-1.51	-1.37

Tabla B.9: Resultados de distribución de raza según su porcentaje de expansión para la instancia I4



## Anexo C

# Validación

### C.1. Evolución del hipervolumen

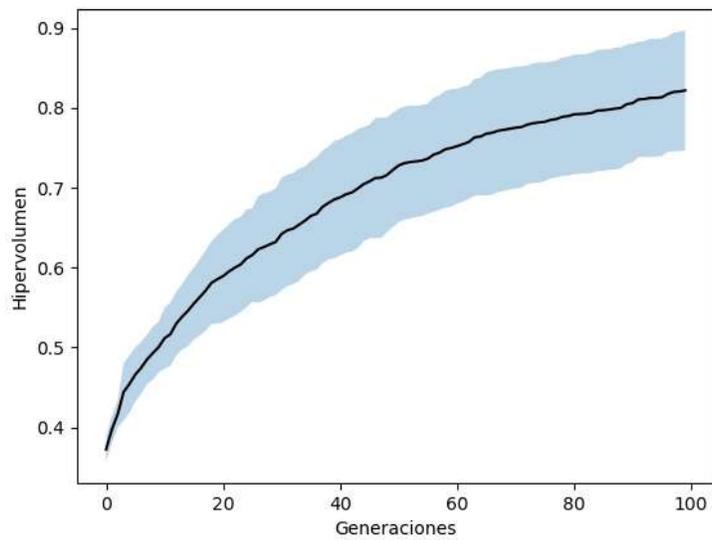


Figura C.1: Evolución del hipervolumen relativo para la instancia I1

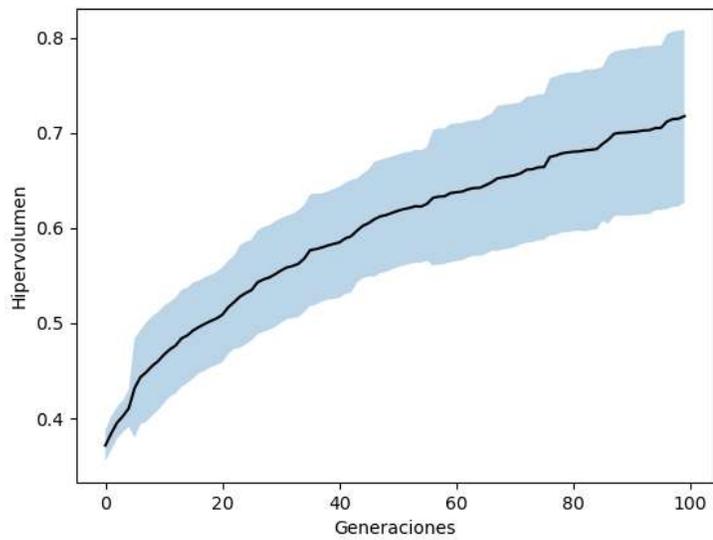


Figura C.2: Evolución del hipervolumen relativo para la instancia I4

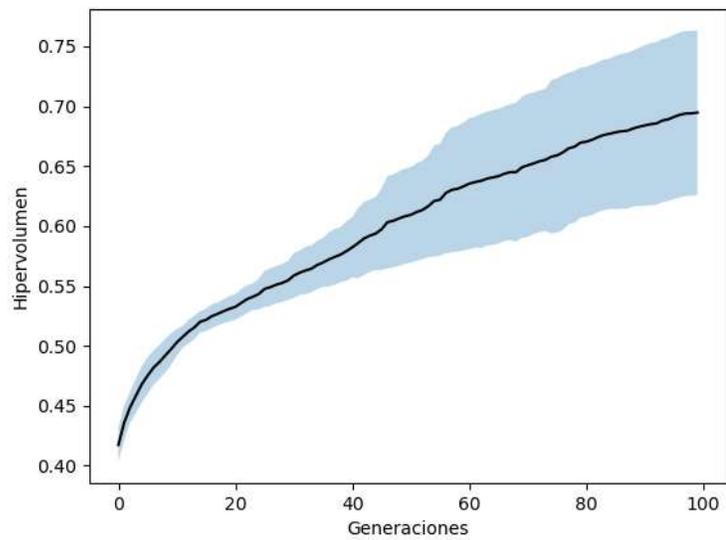


Figura C.3: Evolución del hipervolumen relativo para la instancia I6

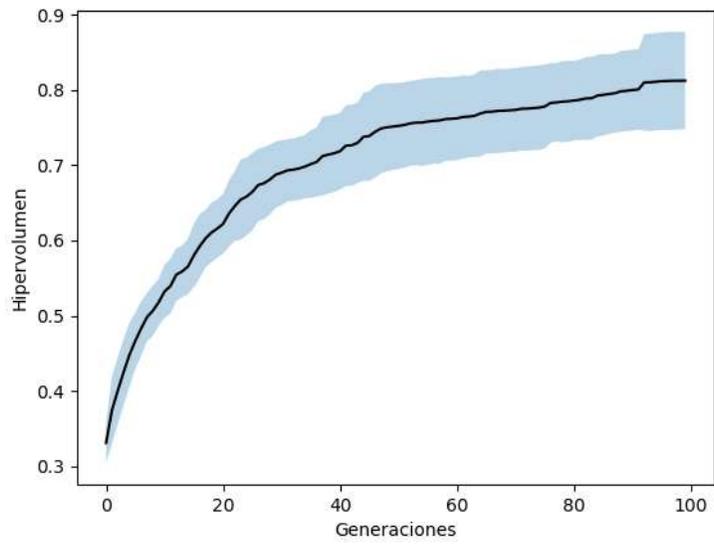


Figura C.4: Evolución del hipervolumen relativo para la instancia I7

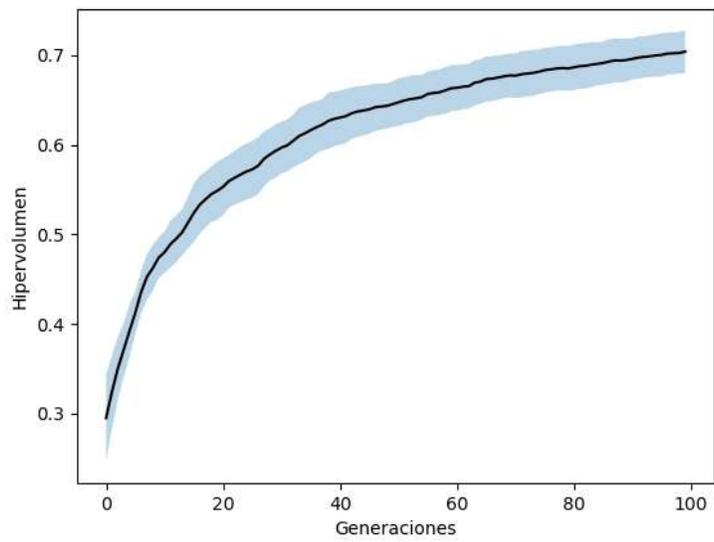


Figura C.5: Evolución del hipervolumen relativo para la instancia I8

## C.2. Evolución del frente de Pareto

### C.2.1. Instancia I1

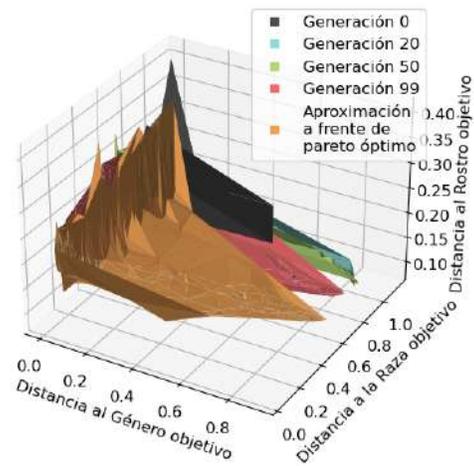


Figura C.6: Evolución del frente de Pareto para la instancia I1

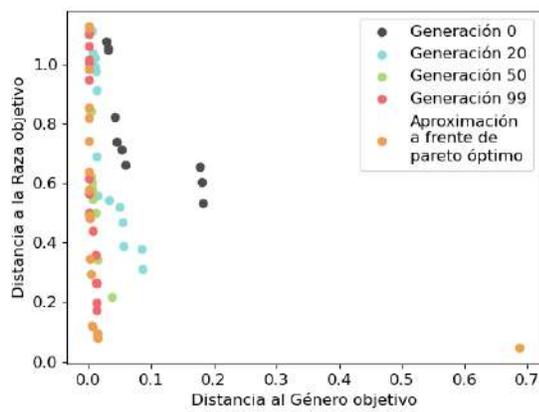


Figura C.7: Corte raza y género: Evolución del frente de Pareto para la instancia I1

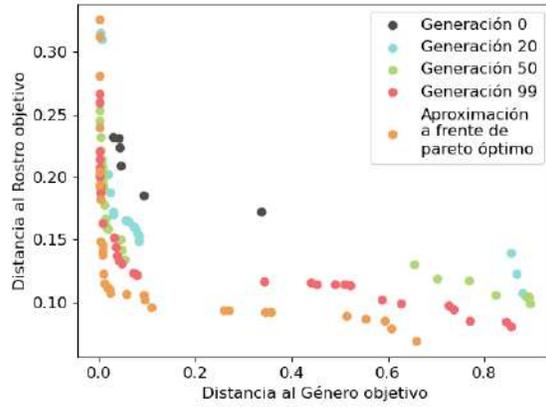


Figura C.8: Corte similitud y género: Evolución del frente de Pareto para la instancia II

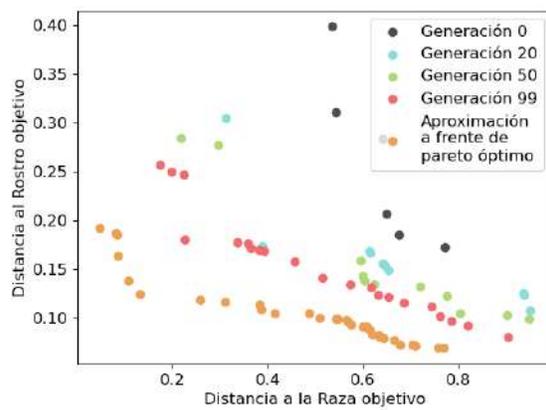


Figura C.9: Corte similitud y raza: Evolución del frente de Pareto para la instancia II

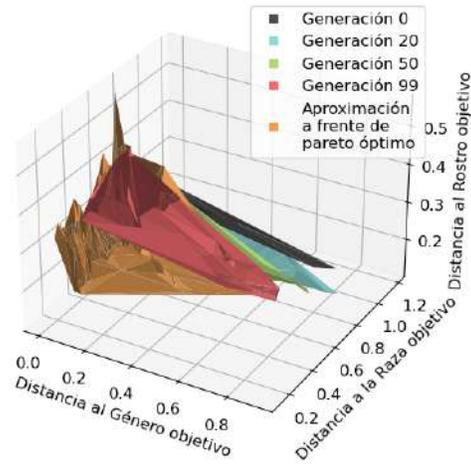


Figura C.10: Evolución del frente de Pareto para la instancia I4

### C.2.2. Instancia I4

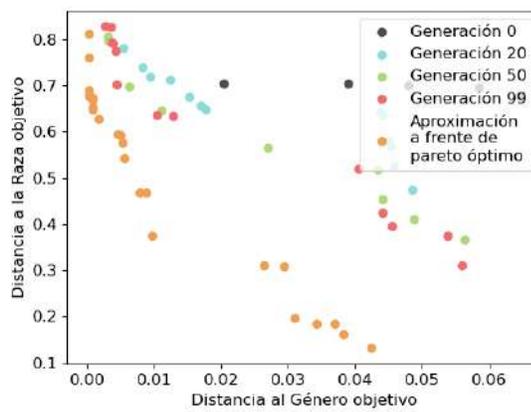


Figura C.11: Corte raza y género: Evolución del frente de Pareto para la instancia I4

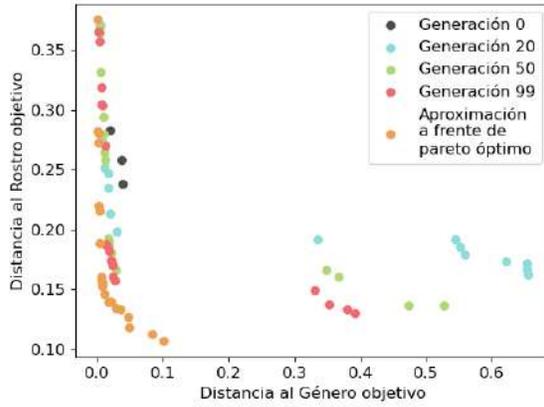


Figura C.12: Corte similitud y género: Evolución del frente de Pareto para la instancia I4

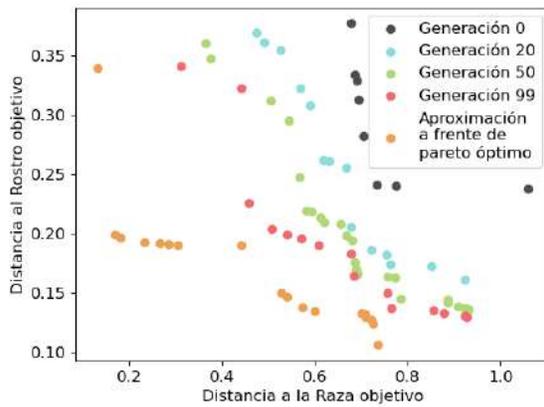


Figura C.13: Corte similitud y raza: Evolución del frente de Pareto para la instancia I4

### C.2.3. Instancia I6

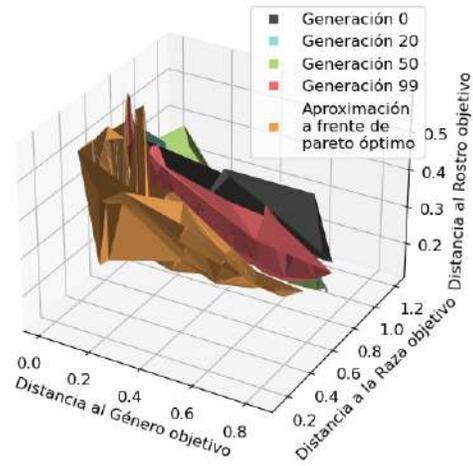


Figura C.14: Evolución del frente de Pareto para la instancia I6

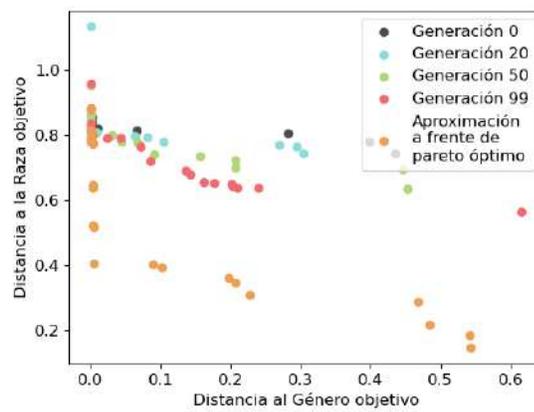


Figura C.15: Corte raza y género: Evolución del frente de Pareto para la instancia I6

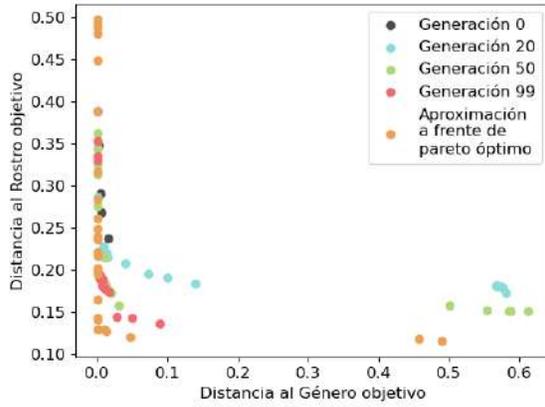


Figura C.16: Corte similitud y género: Evolución del frente de Pareto para la instancia I6

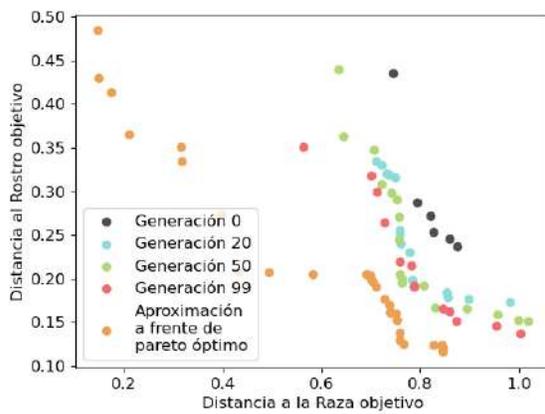


Figura C.17: Corte similitud y raza: Evolución del frente de Pareto para la instancia I6

### C.2.4. Instancia I7

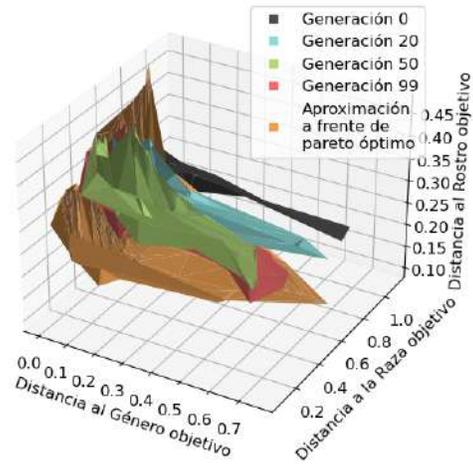


Figura C.18: Evolución del frente de Pareto para la instancia I7

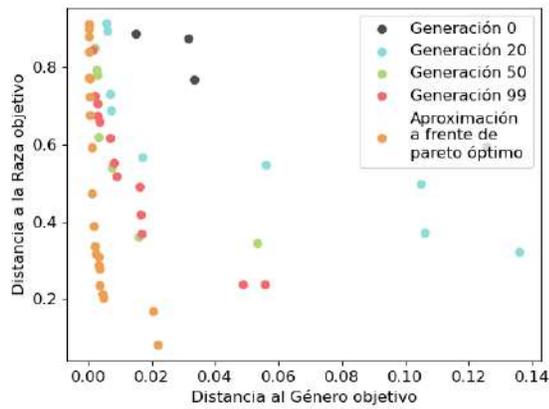


Figura C.19: Corte raza y género: Evolución del frente de Pareto para la instancia I7

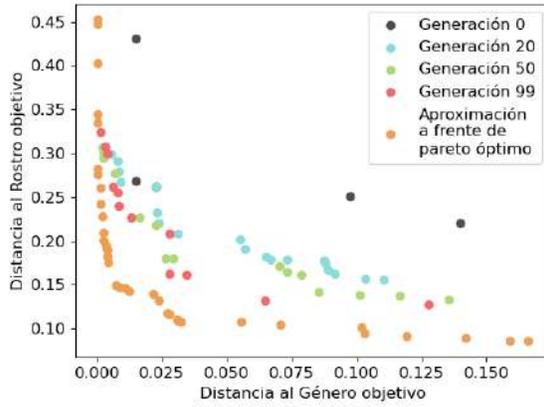


Figura C.20: Corte similitud y género: Evolución del frente de Pareto para la instancia I7

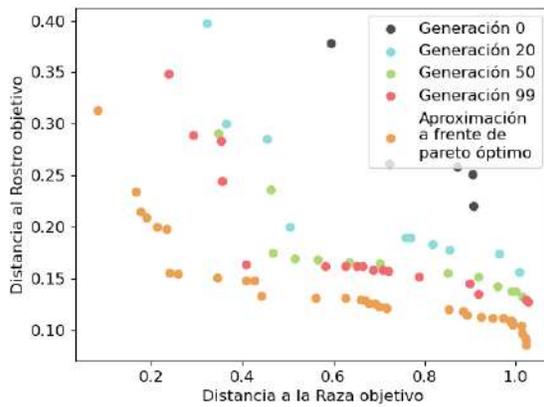


Figura C.21: Corte similitud y raza: Evolución del frente de Pareto para la instancia I7

### C.2.5. Instancia I8

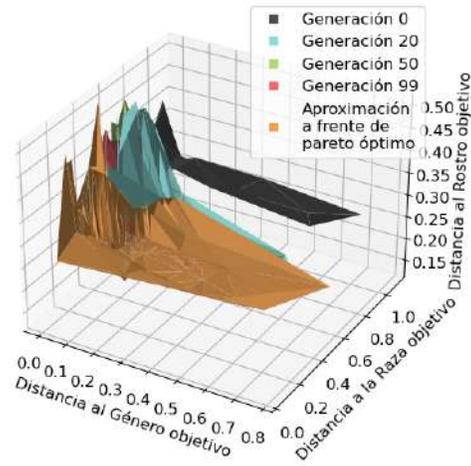


Figura C.22: Evolución del frente de Pareto para la instancia I8

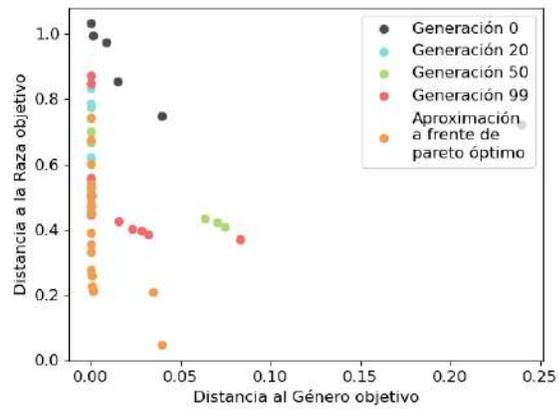


Figura C.23: Corte raza y género: Evolución del frente de Pareto para la instancia I8

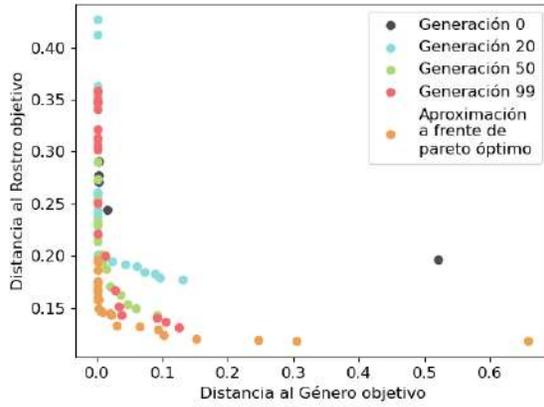


Figura C.24: Corte similitud y género: Evolución del frente de Pareto para la instancia I8

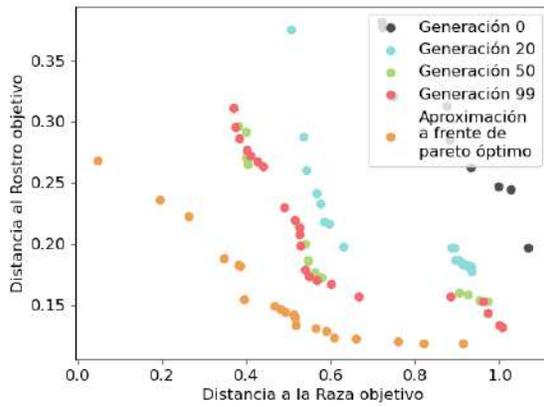


Figura C.25: Corte similitud y raza: Evolución del frente de Pareto para la instancia I8