

Polidata: Modelo de inteligencia artificial para la evaluación del diseño de interfaz de usuario en dispositivos móviles

Polidata: Artificial intelligence model for the evaluation of user interface design on mobile devices

 **José Luis Santorcuato-Tapia**
Universitat Politècnica de València
jlsantap@upv.edu.es

 **Jaime Lloret**
Universitat Politècnica de València
jlloret@ddcom.upv.es

Resumen

Dada la proliferación de *chats* y *prompts* para el uso de inteligencia artificial, especialmente en el ámbito del diseño y generación de contenidos, aparecen desafíos y oportunidades para el diseño, no solo para las etapas de producción o generación de contenidos, también, para las etapas de prototipado y validación de prototipos. Este artículo propone la investigación y el desarrollo de un modelo de aprendizaje profundo multiclasa, con la finalidad de evaluar el diseño de interfaz de usuario para dispositivos móviles, especialmente como apoyo durante el proceso de prototipado de interfaces en alta fidelidad. También, presenta la creación de un *dataset* o conjunto de datos destinado a ser la base de conocimiento del modelo. Los ejemplos que constituyen este *dataset* provienen de aplicaciones disponibles para el sistema Android, empleando, como criterio de selección, el sistema de diseño *Material Design*, propuesto por Google, tomando en consideración sus especificaciones en el uso de tipografías, colores, formas, tamaños, elevaciones, proporción y diagramación. El modelo

Artículo original/*Original Article*

Correspondencia / *Correspondence*
jlsantap@upv.edu.es

Financiación / *Fundings*:
Sin financiación

Recibido / *Received*: 30/09/2023
Aceptado / *Accepted*: 19/12/2023
Publicado / *Publicado*: 28/12/2023

Cómo citar este trabajo.

How to cite this paper:
Santorcuato-Tapia, J. L. y Lloret, J. (2023). Polidata: Modelo de inteligencia artificial para la evaluación del diseño de interfaz de usuario en dispositivos móviles. I+Diseño. Revista Científica de Investigación y Desarrollo en Diseño, 18.

DOI: 10.24310/
idiseo.18.2023.17687

se ha integrado a una interfaz que permite de manera intuitiva obtener inferencias o el nivel de confianza frente a una interfaz diseñada que se le presente al modelo, otorgando una evaluación o valoración de acuerdo con las características que el modelo ha aprendido durante el proceso de entrenamiento. Actualmente el modelo realiza una evaluación cuantitativa, para la que se proponen métricas de evaluación.

Palabras clave: Interfaz gráfica de usuario, diseño, inteligencia artificial, prototipos, evaluación.

Abstract

Given the proliferation of chats and prompts for the use of artificial intelligence, especially in the field of design and content generation, challenges and opportunities for design appear, not only for the stages of production or content generation, but also for the stages of prototyping and validation of prototypes. This article proposes the research and development of a deep learning multiclass model, aimed at evaluating the design of the user interface for mobile devices, especially as support during the prototyping process of high fidelity interfaces. It also presents the creation of a dataset intended to be the basis of knowledge of the model. The examples that constitute this dataset come from applications available for the Android system, using, as a selection criterion, the Material Design design system, proposed by Google, taking into consideration its specifications in the use of typography, colors, shapes, sizes, elevations, proportion and diagramming.

The model has been integrated into an interface that allows intuitively to obtain inferences or the level of confidence in front of a designed interface that is presented to the model, granting an evaluation according to the characteristics that the model has learned during the training process. Currently the model carries out a quantitative evaluation, for which evaluation metrics are proposed.

Keywords: Graphical user interface, design, artificial intelligence, prototypes, evaluation

1. Introducción

La aparición de herramientas de inteligencia artificial para el diseño y la generación de contenidos, entre otras, DALL-E (OpenAI, 2022-a), *Stable Diffusion* (Stability AI, 2022), y *Chat GPT* (OpenAI, 2022-b), han presentado un nuevo escenario para la industria de las comunicaciones, especialmente para la generación de contenidos. En el ámbito específico del diseño, *Photoshop* ya cuenta con una herramienta para la edición y creación de imágenes (Adobe, 2023-a). *Illustrator* también posee un plugin o módulo para colorear o restaurar imágenes (Adobe, 2023-b), al igual que *Premiere* para la edición audiovisual o videográfica (Adobe, 2023-c). *Canva*, la herramienta de diseño gráfico también incluye la posibilidad de integrar la generación automática de contenidos a partir de entradas de texto (Canva, 2023). Para esta investigación resulta relevante la integración de herramientas de inteligencia artificial en *Figma* (Figma, 2015), por ser la herramienta principal para el diseño de interfaz y prototipos de aplicaciones. *Figma* cuenta con una extensión para la creación de íconos SVG, generación de texto y *wireframes*¹. Sin embargo, pese a lo novedoso y útil de la aplicación de la inteligencia artificial generativa, es solo una de las posibilidades de aplicación. La inteligencia artificial puede ser utilizada en diversas etapas del proceso de diseño, como un agente colaborador en las etapas de validación o evaluación de prototipos, que es lo que se propone en esta investigación y proyecto. El diseño de

1. Bocetos preliminares de una interfaz.

una *dataset*² y entrenamiento de un modelo de inteligencia artificial, no es una tarea sencilla, a pesar de que hoy existen plataformas denominadas *no code*³ o *low code*, como *Lobe* (Lobe, s-f) o *Google Cloud Auto ML* (Google Cloud, s-f), entre otras herramientas que facilitan el proceso de creación de datasets, etiquetado de datos o el entrenamiento de modelos de visión computarizada, sin embargo, el desarrollo de herramientas personalizadas de inteligencia artificial, para tareas complejas o problemas específicos, como podrían ser la detección de objetos o segmentación de objetos complejos, muchas veces no poseen modelos pre entrenados, *datasets*, o no permiten, por ejemplo, ajustes de hiper parámetros o personalización de las redes. Por lo anterior, sigue siendo una tarea compleja la integración de herramientas de IA⁴ para la mayoría de los profesionales que no tienen estudios de programación, *Machine Learning*⁵ o conocimientos informáticos avanzados, cuando el problema no es genérico. Algo similar ocurre con herramientas como *Wordpress* (Wordpress, s-f) para el desarrollo web o con *Construct 3* (Construct, s-f) para el desarrollo de videojuegos; en el momento que se requiere integrar con otra plataforma, desarrollar una nueva funcionalidad o se necesita modificar en detalle algún parámetro, que no es parte del ecosistema o de las funcionalidades del *framework*, no se puede o se complejiza en demasía.

La construcción o diseño de una interfaz de usuario obedece a dar solución a un problema en un escenario en particular. Es un diálogo entre una máquina y un usuario en un determinado contexto, o al menos una mimesis de una conversación (Tidwell et al., 2020). El diseño de interfaz de usuario es la culminación de un proceso de investigación, en un terreno específico, en que el diseño debe plasmar sus hallazgos, obtenidos mediante herramientas cualitativas y cuantitativas, como propone Kalbach a lo largo de su libro *Mapping experiences: A Complete Guide to Creating Value Through Journeys, Blueprints, and Diagrams* (2016). En vista de lo anterior expuesto, debería ser considerada, la inteligencia artificial, como una entidad colaboradora en el proceso de diseño, no solo como una herramienta que toma decisiones o es capaz de generar contenidos o prototipos, más bien, como una herramienta que ayuda o facilita la toma de decisiones en las diferentes etapas del proceso de diseño, colaborando en el análisis de las propuestas de diseño y transformándose en una herramienta más, disponible para el proceso de diseño.

La industria y las escuelas de diseño ya comienzan a discutir sobre el impacto de la inteligencia artificial, tanto en el proceso de producción, como en el de enseñanza. El XIV Congreso latinoamericano de enseñanza del diseño, celebrado en julio del año 2023, el que reúne al Foro de escuelas de diseño de Iberoamérica, tuvo como tema central la inteligencia artificial. Se presentaron problemáticas vinculadas con la creación, autoría, ética y evaluación. En la ponencia *Evaluación del diseño de interfaces de usuario con inteligencia artificial* (Santorcuato-Tapia & Lloret, 2023), se discutieron escenarios y desafíos del diseño mediado por inteligencia artificial, expuesta como un agente facilitador en el proceso de enseñanza del diseño digital, a partir de las capacidades de las redes neuronales artificiales en el análisis de imágenes.

La inteligencia artificial puede ser utilizada en diversas etapas del proceso de diseño, como un agente colaborador en las etapas de validación o evaluación de prototipos.

2. Conjunto de datos.
3. Herramientas que no necesitan código o tienen un sistema de programación visual.
4. Inteligencia artificial.
5. Aprendizaje de máquina.

La industria y las escuelas de diseño ya comienzan a discutir sobre el impacto de la inteligencia artificial, tanto en el proceso de producción, como en el de enseñanza.

La mayoría de los modelos de inteligencia artificial disponibles para usuarios finales⁶ son redes generativas, basadas en *GANS*⁷ o conocidas en español como redes generativas adversarias. Redes orientadas a la generación de objetos, a realizar una mímica de los datos que se le presentan al modelo (Moghadam et al., 2022).

Los objetos generados pueden ser de diferente naturaleza y origen, como modelos 3D, síntesis de voz, imágenes, transferencia de estilos o texturas, con diversos propósitos, orientados a dar soluciones a diferentes disciplinas e industrias (Aggarwal et al., 2021).

En el contexto del diseño de interfaz de usuario la mayoría de las experiencias que no están relacionadas con *GANS*, están circunscritas al campo de la experimentación, de laboratorio, no siendo distribuidas, publicadas o puestas al alcance de los diseñadores, utilizan, además, como se señaló en párrafos anteriores, sistemas complejos, que necesitan una gran cantidad de datos y de recursos informáticos, además de personal técnico altamente cualificado, no disponibles para un usuario final.

Para el problema específico del diseño de interfaces y su evaluación, se plantea, en este artículo y proyecto, la construcción de un *dataset* o conjunto de datos como base de conocimiento de un modelo de aprendizaje profundo, incorporando el estado del arte del diseño de interfaces en dispositivos móviles, primando la calidad de la interfaz por sobre la cantidad de datos o ejemplos disponibles. Para lo anterior, se ha considerado el sistema de diseño de *Google, Material Design 3* (Material Design 3, 2021a). Este sistema de diseño ofrece una guía de buenas prácticas y especificaciones técnicas en cuanto a tipografías, colores, formas, tamaños, elevación, proporción y diagramación. Lo anterior con la finalidad de aumentar la consistencia, accesibilidad y usabilidad de componentes y patrones de diseño.

El objetivo del proyecto es desarrollar un modelo que permita evaluar la calidad de una interfaz, de acuerdo con las dimensiones nombradas en el párrafo anterior, que sirven de guía o elementos para la curatoría en la selección de ejemplos y captura de datos, para posteriormente ser analizados por una red neuronal convolucional, que han demostrado un gran desempeño en tareas de análisis de patrones y su clasificación. Sin embargo, la sola clasificación no es el objetivo de este proyecto, la confianza, expresada en el porcentaje que obtiene la inferencia, es el parámetro principal para la evaluación. Como hipótesis inicial se propuso que el sistema fuera capaz de clasificar las interfaces y asignar un porcentaje de confianza en esas clasificaciones a partir de la calidad de las interfaces y de las características aprendidas durante el proceso de entrenamiento de la red, ese porcentaje de confianza señalaría la calidad de la interfaz, cuán bien o mal está diseñada la interfaz en relación con tipografías, colores, formas, elevaciones, proporción y diagramación. Se hace énfasis en que el modelo propuesto no está diseñado para detectar elementos o componentes, está diseñado para evaluar el diseño de una interfaz determinada, de un patrón de diseño, en su conjunto. Para lo anterior, se proponen métricas que faciliten la interpretación de los resultados, para la discusión de diseñadores y equipos de diseño. También, este proyecto aborda el problema de usabilidad de los sistemas de inteligencia artificial, simplificando la inferencia del modelo con una interfaz gráfica, permitiendo a los usuarios arrastrar o cargar sus imágenes para obtener las inferencias, relevando a los usuarios de tener que instalar sistemas o modificar el código para interactuar con el modelo.

6. Usuarios o consumidores de un producto o servicio.

7. Generative adversarial network.

Con lo anterior se espera promover la adopción del modelo por parte de los usuarios, estudiantes de diseño, diseñadores noveles o profesionales. En general, para quien deba o necesite evaluar diseños propios o propuestas de diseño de aplicaciones móviles, en el sector público o privado.

En la sección 2 se presentan trabajos relacionados. En la sección 3 se incluye el material y método. La sección 4 está compuesta por los resultados y la discusión. Para finalizar, en la sección 5, están las conclusiones y las próximas etapas del proyecto.

2. Trabajos relacionados

A continuación, se revisarán parte los trabajos más relevantes relacionados con los *datasets* orientados al diseño de interfaz de usuario y su utilización en diferentes modelos propuestos por otros autores, así también, una revisión de la problemática en la evaluación del diseño de interfaz de usuario y sistemas de diseño.

2.1. Datasets y modelos de inteligencia artificial

Existen numerosos *datasets* construidos para el problema del diseño de interfaz de usuario, es el caso de *Erica* (Deka et al., 2016), *Enrico* (Leiva et al., 2020), *UI5K* (Chen et al., 2021), *WebUI* (Wu et al., 2023), sin embargo, la mayoría se ocupa del volumen de datos por sobre la calidad de ellos, están destinados a ser la base de conocimiento de modelos generativos o tienen por objetivo la programación de interfaces.

El *dataset RICO* (Deka et al., 2017) es probablemente el *dataset* más utilizado y citado en las investigaciones y aplicaciones de *Machine learning* en el problema del diseño de interfaces, es el que presenta mayor volumen de datos y es un buen referente para comprender el cómo diseñar un *dataset* para enfrentar el problema del diseño de interfaz de usuario, además de ofrecer una amplia documentación relacionada con el proceso de entrenamiento. *RICO* está construido para el análisis de interfaces de usuario en aplicaciones móviles mediante la utilización de herramientas de *Machine learning*, presentado el año 2017, incluye 9.300 aplicaciones analizadas, 66.000 imágenes únicas y 27 categorías.

RICO presenta 5 modelos:

- Búsqueda de diseño, por similitud.
- Generación de interfaces de usuario.
- Generación de código, Java-Android.
- Modelado de interacción.
- Predicción de percepción de usuario.

El ámbito de aplicación de la inteligencia artificial aplicada al diseño de interfaz de usuario sigue siendo, en su mayoría, la generación de interfaces, o la búsqueda de similitud de una interfaz con alguna presente en el *dataset*. Es el caso de *Screen2Vec* (Li et al., 2021) basado en *Word2Vec* (Mikolov et al., 2013). Es un modelo que a partir del *dataset RICO*, obtiene *embeddings* o representaciones vectoriales, logrando en su salida la reconstrucción de la interfaz solicitada mediante un texto de entrada.

ScreenQA se propone como modelo basado en *VQA*⁸ (Srivastava et al., 2020). También, utiliza el *dataset RICO*, permitiendo la búsqueda de elementos, componentes y estructuras, a partir de una entrada de texto. Se realizaron más de 80.0000 pares de anotaciones, texto vs. imágenes. Este modelo genera interfaces a nivel *embeddings*,

El ámbito de aplicación de la inteligencia artificial aplicada al diseño de interfaz de usuario sigue siendo, en su mayoría, la generación de interfaces, o la búsqueda de similitud de una interfaz con alguna presente en el *dataset*.

8. Visual question answering.

sin incluir propiamente el diseño de interfaz por parte del diseñador, solo genera plantillas (Hsiao et al., 2022).

Keselj et al. (2022), proponen un paradigma híbrido de recomendación de interfaz de usuario para su modelo de aprendizaje profundo. Este paradigma se basa en el consumo y preferencias de los usuarios y también en la similitud de las interfaces. Sin embargo, el modelo no realiza evaluaciones de interfaces en alta fidelidad que el diseñador proponga o haya diseñado.

Una aproximación a este proyecto e investigación es *UIED* (UIED, 2020) basado en el artículo de Xie et al., (2020), una plataforma que permite la carga de diseños de interfaz de usuario y su posterior análisis, sin embargo, está orientada al rediseño visual de los elementos de la interfaz de usuario detectados, su reorganización en términos de diagramación y etiquetado, entre otros elementos a modificar, no otorgando una evaluación o porcentaje de confianza, tampoco realiza recomendaciones técnicas, se orienta, además, a la generación de código.

Otro trabajo, en una línea similar a la presente investigación y proyecto, es presentado por Altinbas et al., (2022) en el que integran *Yolo V5* (Ultralytics, 2020) para detección de elementos de la interfaz de usuario, sin embargo, nuevamente el enfoque es la detección de elementos de la interfaz de usuario.

La mayoría de los trabajos anteriores tienen una orientación hacia la generación de interfaces, detección de elementos, desarrollo de interfaces, software o aplicativos, sin mucho énfasis en la calidad del diseño de interfaz de usuario o su evaluación, además, incluyen, una terminología demasiado técnica para un diseñador de experiencia de usuario o interfaces.

También, presentan una complejidad en la implementación, que requiere de equipos de ingenieros en inteligencia artificial para su uso y mantenimiento, como se ha expresado de anteriormente.

2.2. Diseño y evaluación de interfaz de usuario

Garrett (2011) propone 5 planos para el diseño de experiencia de usuario en la web, que han sido adoptados y utilizados en el diseño de aplicaciones móviles. Lo constituyen la definición de objetivos que se pretenden alcanzar, investigación de elementos etnográficos, casos de uso, comparativas, arquitectura de la información, diseño de interacción, navegación, diseño de interfaz y diseño visual. Por tanto, el diseño de interfaz de usuario es el resultado de todo el proceso de diseño de experiencia de usuario.

En el proceso de diseño, específicamente en el diseño de interfaz gráfica de usuario, la etapa de evaluación de prototipos ha utilizado tradicionalmente dos herramientas: la evaluación heurística y las pruebas con usuarios, ambas herramientas son ampliamente utilizadas por la comunidad y profesionales del diseño de experiencia de usuario.

En su artículo *Enhancing the explanatory power of usability heuristics*, Jakob Nielsen, presenta una profundización sobre la heurística y su aplicación, plantea 7 factores para evaluar un sistema en relación con la usabilidad o problemas de usabilidad (Nielsen, 1994). Estos factores son:

- Visibilidad del estado del sistema.
- Relación entre el sistema y el mundo real.
- Control por parte del usuario y libertad.
- Consistencia y estándares.
- Prevención de errores.
- Reconocimiento antes que el recuerdo.
- Flexibilidad y eficiencia de uso.

En el proceso de diseño, específicamente en el diseño de interfaz gráfica de usuario, la etapa de evaluación de prototipos ha utilizado tradicionalmente dos herramientas: la evaluación heurística y las pruebas con usuarios.

En el caso del diseño de interfaz de usuario los factores más relevantes resultan ser: consistencia y estándares, reconocimiento antes que el recuerdo.

Desde el punto de vista morfológico y del diseño de interfaz de usuario, el primero está relacionado con los sistemas de diseño, la construcción de elementos, componentes y patrones de diseño a lo largo de un sistema o aplicación.

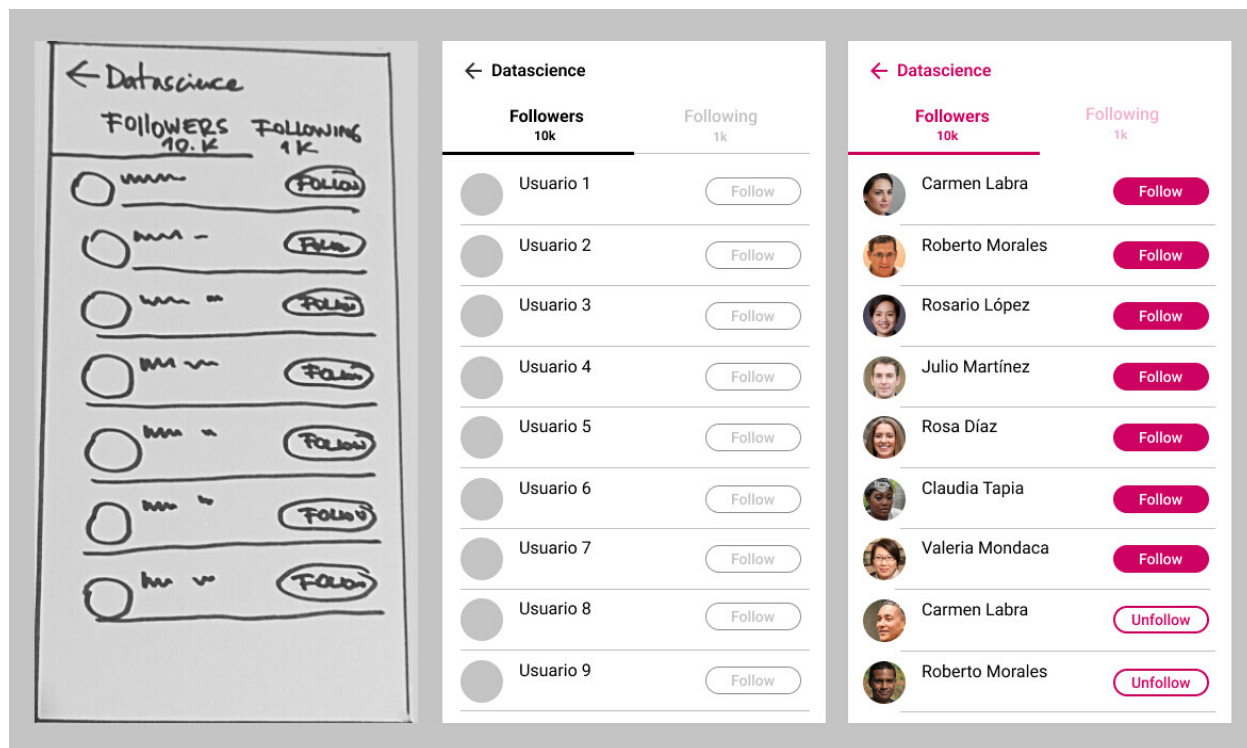
El segundo factor, no excluyente, está vinculado al reconocimiento de patrones, acciones, y, por tanto, disminuir la carga de memoria del usuario, a partir de su experiencia previa con aplicativos.

La prueba o evaluación heurística requiere de un equipo de diseñadores de UX⁹ expertos, para, mediante una matriz, evaluar las diferentes dimensiones de un aplicativo, básicamente su impacto en la usabilidad, que incluye el diseño de interfaz de usuario y los factores planteados por Nielsen ya mencionados.

En las pruebas con usuarios se evalúa el desempeño de una interfaz con usuarios, puede ser en etapas tempranas, utilizando *wireframes*, en etapas más avanzadas, con interfaces en una fidelidad media y alta. Las pruebas con usuarios se realizan mediante la asignación de tareas a un usuario, medición del tiempo que le toma a los usuarios completarlas, porcentaje de cumplimiento de ellas y una entrevista cualitativa que permite evaluar la experiencia del usuario y eventuales problemas de usabilidad o experiencia. Ambas herramientas, tanto la evaluación heurística, como las pruebas con usuarios, entregan hallazgos que permiten obtener aprendizajes para una siguiente iteración de los prototipos.

En la Figura 1 se puede ver las distintas etapas del proceso de diseño de una interfaz de usuario.

Figura 1. Etapas del prototipo de interfaz de usuario, de izquierda a derecha: baja fidelidad, media fidelidad, alta fidelidad. Avatares de usuario generados con IA, plugin Avatar Generator. Fuente: Figma, 2022.



9. Experiencia de usuario.

Las pruebas señaladas en párrafos anteriores son situaciones en las que el estudiante o el profesional novel del diseño de interfaces no posee la experiencia para validar o descartar sus diseños, dado que las pruebas son aplicadas por expertos.

Los estudiantes y profesionales del diseño deben, principalmente, para la construcción de interfaces, observar interfaces de otros aplicativos o sistemas, copiarlas y modificarlas, obteniendo retroalimentación de sus colegas, docentes, clientes o del equipo encargado de las diferentes pruebas.

El aprendizaje de máquina requiere *datasets* con gran cantidad de ejemplos. El artículo *How many images do I need?* de Shahinfar et al., (2020), presenta diversas experiencias con conjunto de datos balanceados¹⁰ con diferentes cantidades de datos, desde 10 imágenes, hasta 1.000 imágenes por clase o categoría a clasificar. La eficiencia de las redes ha avanzado mucho en los últimos 5 años, así también, la capacidad de cómputo de los ordenadores.

En este problema, en particular, para el análisis de interfaces, los datos son las capturas de pantalla de diferentes aplicaciones móviles, además, estos datos, a veces disponibles, no siempre están actualizados frente a las tendencias del diseño de aplicaciones que evoluciona rápidamente. También, muchas aplicaciones protegen sus productos con *DRM*¹¹, lo que no permite realizar capturas de pantalla o contar con todos los datos necesarios para construir un conjunto de datos explotable.

«Los sistemas de diseño son un lenguaje compartido sobre el cual se generará la comunicación al interior de una organización o equipo de diseño».

2.3. Sistemas de diseño, Material Design 3

Los sistemas de diseño son, en palabras de Vesselov y Davis (2019, pp. 51-62), un lenguaje compartido sobre el cual se generará la comunicación al interior de una organización o equipo de diseño. Están constituidos por una guía de estilos compuesta por la familia de tipografías a utilizar, sus tamaños y pesos. Incluye colores primarios, secundarios, colores de acento y neutros. También, incluye el diseño de elementos, componentes, elevaciones, iconografía y patrones de diseño. La utilización de un sistema de diseño garantiza consistencia a lo largo de la experiencia del usuario, al estar diseñados previamente todos los elementos de una interfaz. Por lo anterior, durante su despliegue mantiene consistencia en el lenguaje visual, permitiendo que los usuarios identifiquen elementos estáticos, interactivos, acciones principales y patrones perceptuales y funcionales.

Existen tantos sistemas de diseño como aplicaciones, por ejemplo, los sistemas de diseño *Base* de Uber (Uber, 2023), *Carbon* de IBM (IBM, s-f), *Fluent UI* de Microsoft (Microsoft, s-f), entre muchos otros.

Para esta investigación y proyecto se ha seleccionado *Material Design 3* que es el sistema de diseño de Google para sus aplicaciones, tanto de escritorio como móvil. Está presente en todas las aplicaciones que la compañía ofrece a los usuarios: *Drive*, *Gmail*, *Maps* y *Youtube*, entre otras. Se ha seleccionado por la extensa documentación que ofrece y por ser un estándar en la industria del diseño digital. *Material Design 3* ofrece en su sitio una guía completa de los elementos que constituyen un sistema de diseño, para su aplicación por parte de los diseñadores. Entregando, además, especificaciones técnicas y morfológicas para su implementación práctica en la etapa de diseño de interfaz de usuario y en las etapas de desarrollo de software, presentando ejemplos de codificación para lenguajes como *Flutter* (Flutter, s-f) y *web*, entre otros.

10. Misma cantidad de ejemplos por clase o categoría a clasificar o evaluar.

11. Digital Rights Management, gestión de derechos digitales

A continuación, se presentan parte de la guía de estilos y ejemplos del sistema de diseño *Material Design 3* para su aplicación en el diseño de una interfaz de usuario en dispositivos móviles.

En la Figura 2 se observa la definición de tipografías, tamaños propuestos para el diseño de interfaces por *Material Design 3*.

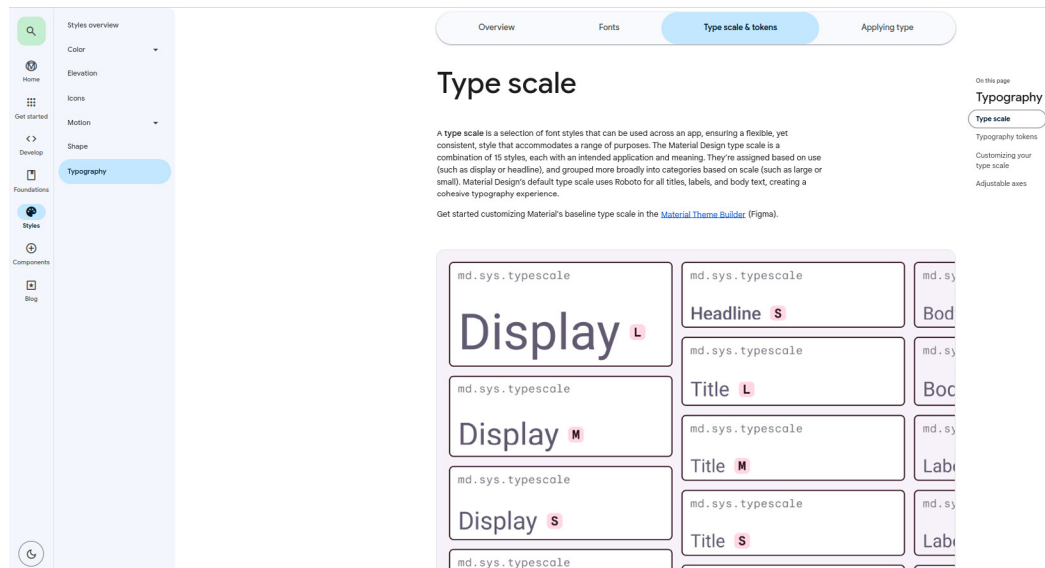


Figura 2. Especificación de tamaños de tipografía y pesos, para diferentes elementos en la interfaz de usuario (Material Design 3, 2021b).

En la Figura 3 se observa un ejemplo acerca de la definición de colores en *Material Design 3*, en la que destacan primario, secundario, terciario y neutrales.

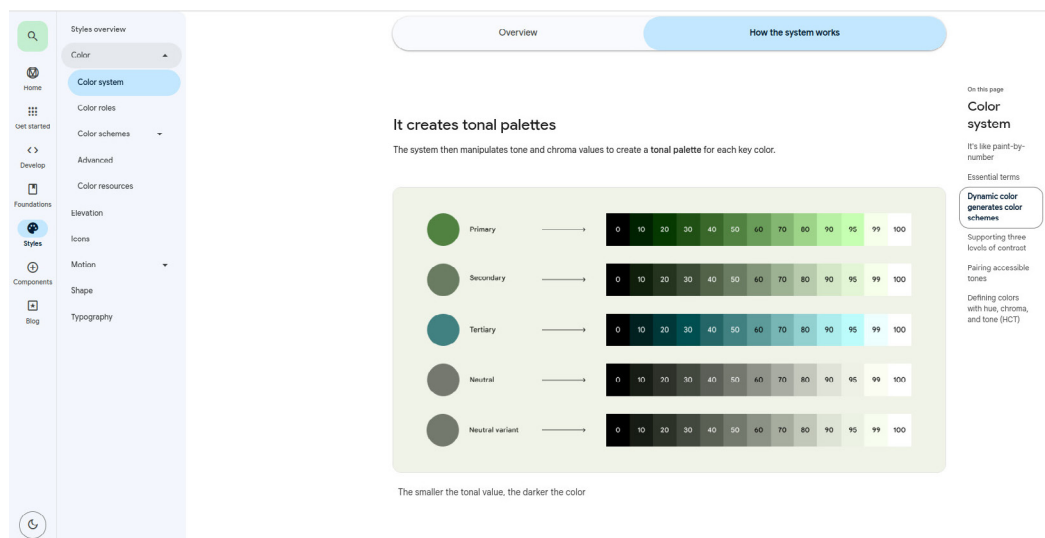


Figura 3. Especificación de colores primarios, secundarios, terciarios y neutros, también su graduación en términos de opacidad (Material Design 3, 2021c).

En la Figura 4 se la guía de especificaciones técnicas propuestas por *Material Design 3*, para el componente *list*, en términos de elementos que lo componen y diagramación.

En la Figura 5 se presentan las dimensiones o tamaños expresados en píxeles propuestas por *Material Design 3* para el componente *list*.

En la figura 6 se presenta el uso del componente *list* propuesto por *Material Design 3* para una interfaz.

Figura 4. Anatomía de elementos que constituyen el componente list (Material Design 3, 2021d).

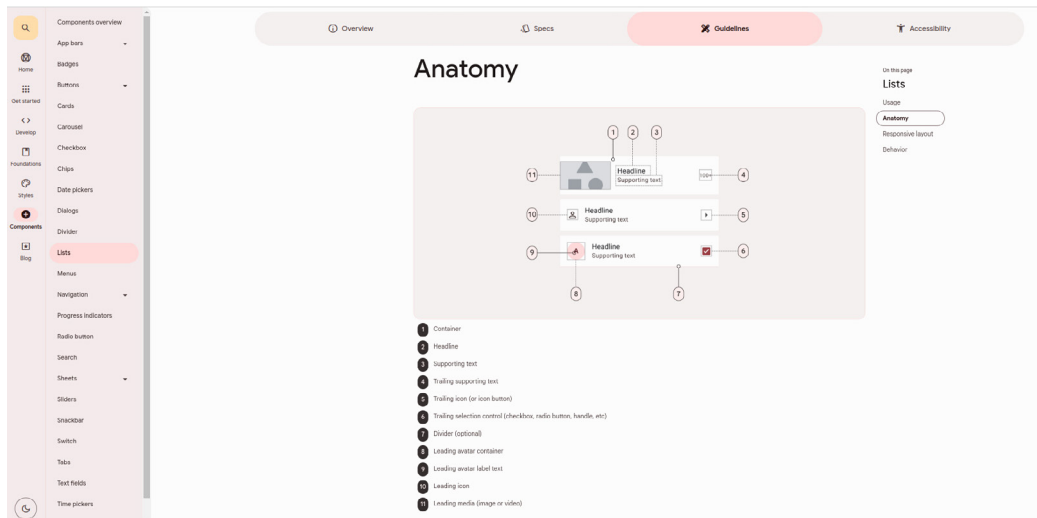


Figura 5. Medidas o tamaños de los elementos del componente list, expresado en píxeles, generalmente múltiplos de 8 píxeles (Material Design 3, 2021e).

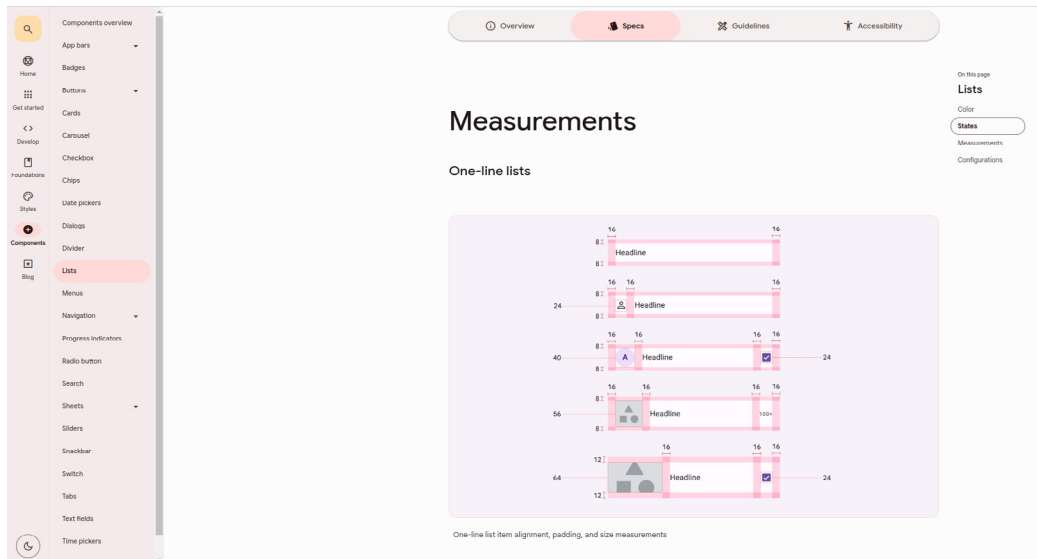
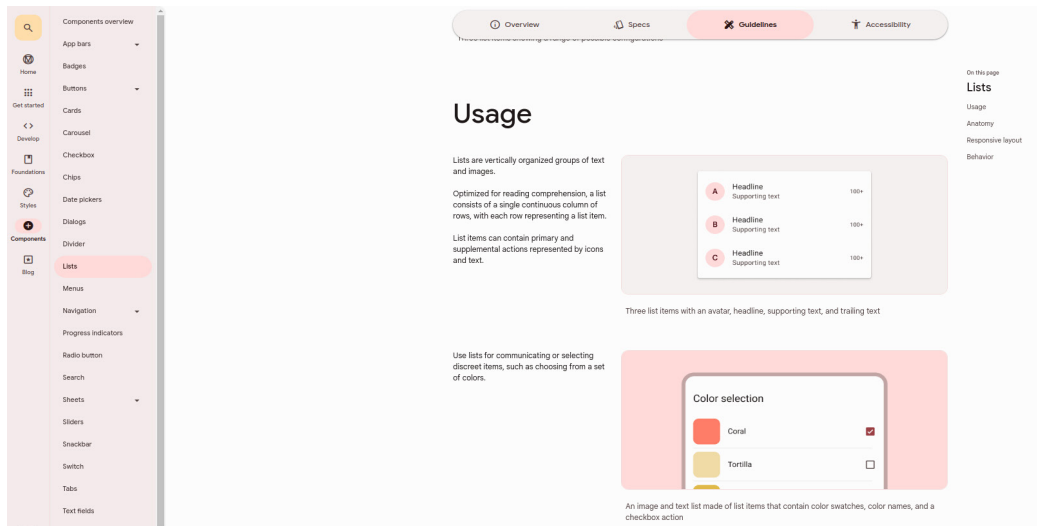


Figura 6. Uso propuesto del componente list por Material Design 3, integrado en una interfaz de usuario en dispositivos móviles (Material Design 3, 2023f).



2.4. Patrones de diseño

Frost, en su libro *Atomic Design* (2016) realiza una analogía entre la química y el diseño. Plantea la similitud desde los átomos, que son indivisibles; la unidad más pequeña y que, agrupados, generan moléculas y posteriormente organismos. El planteamiento de Frost es exclusivamente desde el diseño de interfaz de usuario, dejando de lado cualquier mención a código. En el caso del diseño, estos átomos son, por ejemplo, los botones, campos de textos, etiquetas e íconos, es decir, los elementos más pequeños de una interfaz, que, combinados, generan unidades de significado y funcionalidad, el planteamiento de Frost es similar al que posteriormente realizan Vasselov y Davis (2019, p. 54). Estas moléculas, agrupadas de diferente manera dan paso a componentes que pueden servir para enviar datos de un formulario, buscar un archivo, seleccionar una fecha, entre otras funcionalidades presentes en un aplicativo. Esta evolución de átomos a moléculas, da paso a los organismos, plantillas y pantallas, la interfaz gráfica de usuario, que será interpretada por los usuarios al reconocer los patrones de diseño y su funcionalidades, pero también presenta un problema a la hora de su análisis mediante herramienta de *Machine learning*: los elementos, las unidades más pequeñas, son siempre las mismas, pueden repetirse en una serie de patrones, su combinación da paso a los diferentes organismos y su evolución, por ejemplo, una *card* o tarjeta, puede tener un botón de acción, al igual que un formulario, o un reproductor de vídeo tendrá elementos similares a un reproductor de música, botones de reproducción, pausa, rebobinado, entre otros, complejizando el aprendizaje de máquina, necesitando de muchos más datos o ejemplos para comprender el contexto de uso de la interfaz presentada.

Kholmatova (2017), define los patrones de diseño, como patrones perceptuales y funcionales, aquella organización de elementos o componentes en una interfaz que son percibidos y reconocidos a lo largo del diseño de una aplicación o sistema. El primero se ocupa de la utilidad y uso de una interfaz al usuario, a partir del propio mapa mental y experiencia. El segundo, de su morfología, es decir, ubicación espacial de los elementos o componentes, tamaños, colores y textos que refuerzan los contextos y acciones. Esta organización le permite al usuario comprender el uso y funcionalidad de una interfaz.

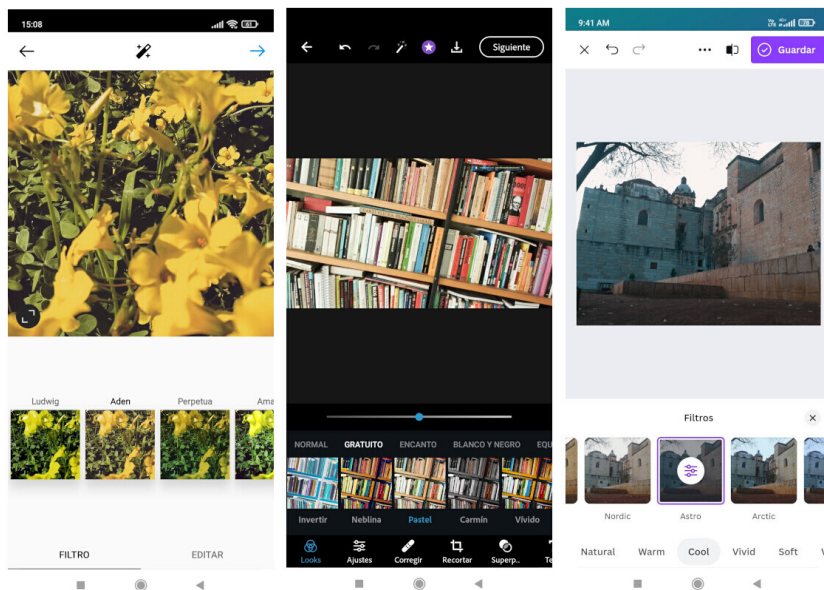


Figura 7. Patrón de diseño para aplicación de filtros a una imagen en Instagram, Photoshop y Canva en dispositivos móviles con sistema Android.

En la Figura 7 se aprecia un ejemplo de patrón de diseño, un filtro para aplicar a una imagen, en 3 aplicaciones móviles en el sistema *Android*: *Instagram*, *Photoshop* y *Canva*. En la Figura 8 se presenta el patrón de diseño de aplicaciones de *streaming* de música, como *Spotify*, *Tidal* y *Apple Music*.

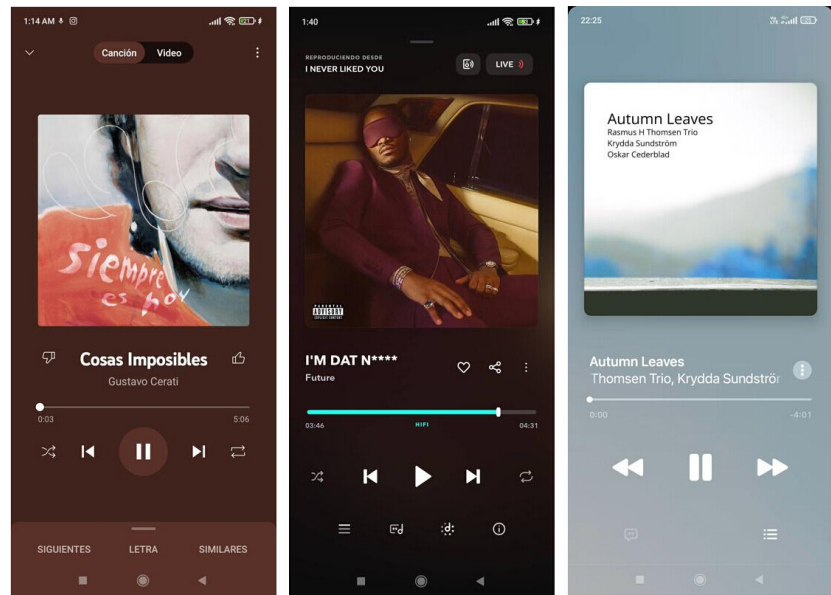


Figura 8.
Patrón de diseño para reproducción de música en Spotify, Tidal y Apple Music en dispositivos móviles con sistema Android.

3. Modelo, redes convolucionales e inferencias

3.1. Datasets

Un *dataset*, es la base de conocimiento sobre el cual se entrenará un modelo de inteligencia artificial, es decir, es el sustrato que permite a los modelos aprender las características o patrones de las imágenes que se le presentan, en este caso, patrones de diseño de interfaces de aplicaciones móviles.

Está constituido por datos no estructurados, imágenes o capturas de pantalla de las aplicaciones.

La problemática de la obtención o volumen de datos es en parte solucionada con aumentación de datos, una técnica que permite, mediante la modificación de las imágenes, aumentar la cantidad de ejemplos disponibles para el entrenamiento.

3.1.1. Características del dataset

El *dataset* es un conjunto de datos multiclase, es decir, permite la clasificación y evaluación de diferentes clases o patrones de diseño, para este caso, 24 patrones de interfaz de usuario.

La recolección de datos, son, en este caso, las capturas de pantallas de diferentes aplicaciones. Estas capturas se realizaron en un terminal *Xiaomi Redmi Note 10 s* con el sistema operativo *Android 11*. Se utiliza para todo el procesamiento y análisis de imágenes un sistema *GNU-Linux Ubuntu 22.0.1* (Canonical, 2022), con el Kernel 5.19.17. Se utiliza, además, una *GPU Nvidia RTX 3060* de 12 GB *VRAM*. El ambiente de desarrollo es *Jupyter* (Project Jupyter, 2014), en su versión 4.0.5, utilizando la versión 3.10.11 de Python, en conjunto con el *framework* de *Machine learning Pytorch 2* (Pytorch, 2023), versión 2.0.1 cu118, de los laboratorios *Meta*.

Se utiliza una carpeta o directorio por cada patrón que se incluye en el *dataset*, al interior de esta carpeta se encuentran todos los ejemplos o capturas que se han recolectado de un determinado patrón.

Para la construcción del *dataset* se analizaron más de 1,200 aplicaciones de manera manual, por inspección visual, de las cuales se obtuvieron datos o capturas de pantalla de 760 aplicaciones que cumplían con las directrices de *Material Design 3*.

Se consideraron para la selección de los ejemplos los tamaños de los elementos y componentes, ubicación de ellos, su elevación, diagramación general de la interfaz, colores para la accesibilidad, iconografía, imágenes, tipografías y la consistencia con otras interfaces o patrones de diseño de uso general por parte de las aplicaciones seleccionadas, como podrían ser un inicio de sesión o un selector de rango de fechas para viajes.

También, se considera la necesidad de obtener al menos 8 ejemplos de un patrón de diseño en una sola aplicación, para reforzar el aprendizaje de un patrón, evitando duplicación de textos, fechas y contenidos, con la finalidad de eludir el aprendizaje de *artefactos*¹² o la memorización de la información.

En la recolección de datos se considera principalmente modo *light* de las aplicaciones, incluye algunas interfaces en modo *dark*, con la finalidad de entregar diversidad a la red, un elemento importante para asegurar una buena generalización por parte del modelo.

La cantidad de ejemplos presentes en el *dataset* es de 9,738 imágenes únicas, sin embargo, no todas las clases o categorías tienen la misma cantidad de ejemplos o imágenes, para lo anterior se decidió balancear el *dataset* a 300 imágenes únicas por clase, de esta manera se evita que una clase que tenga mayor cantidad de ejemplos, presente mayor sesgo o peso frente a otras que tengan menor cantidad de ejemplos.

La división de *sets* o conjuntos de datos se realizó sobre los 300 ejemplos únicos por clase, en la siguiente proporción:

- 80% de ejemplos para el set de entrenamiento, obteniendo 240 ejemplos únicos.
- 10% de ejemplos para el set de validación, obteniendo 30 ejemplos únicos.
- 10% para el set de prueba, obteniendo 30 ejemplos únicos.

Como se trata de un *dataset* compacto, se decide aumentar los datos, mediante cambio de espacio de color, solo para el set de entrenamiento, posterior a la división de *sets*. No se aplicaron otro tipo de aumentaciones como *flip horizontal*, *vertical* o rotación. El contexto de presentación de una interfaz móvil es principalmente retrato, con lectura de izquierda a derecha y de arriba a abajo. Lemley et al., (2017), plantean la aumentación de datos como otra técnica de regularización, en caso de que el *dataset* sea compacto, para evitar sobreajuste¹³.

La aumentación de espacio de color se aplicó a todos los ejemplos presentes en el set de entrenamiento, doblando la cantidad inicial de ejemplos, utilizando la librería *Open CV* (Open CV, s.f.) y su función *COLOR_BGR2RGB* que invierte los canales RGB, intercambiado canales, especialmente útil a partir de la diversidad de colores presentes en los diferentes sistemas de diseño e interfaces. También, se aumentaron en un 30% los datos presentes en el *set* de entrenamiento, cambiando su espacio de

El *dataset* es un conjunto de datos multiclase que permite la clasificación y evaluación de diferentes clases o patrones de diseño.

12. Elementos que no son parte del patrón y que la red puede memorizar, asociándose a un patrón, contaminando la inferencia.

13. Memorización de los datos por sobre el aprendizaje de ellos.

Clase o patrón	Entrenamiento, imágenes aumentadas	Validación, imágenes únicas	Pruebas, imágenes únicas
audio_recorder	552	30	30
card_grid_md	552	30	30
card_grid_sm	552	30	30
card_grid_xl	552	30	30
conversational_ai	552	30	30
crypto	552	30	30
date_range	552	30	30
image_filter	552	30	30
list_md	552	30	30
list_sm	552	30	30
list_profile	552	30	30
list_xl	552	30	30
map	552	30	30
music	552	30	30
nav_drawer	552	30	30
notification	552	30	30
rate	552	30	30
reel	552	30	30
setting	552	30	30
sign	552	30	30
splashscreen	552	30	30
video_fullscreen	552	30	30
walkthrough	552	30	30
weather	552	30	30
Total imágenes	13,248	720	720

Tabla 1.

Número de clases, nombre de clases y ejemplos presentes en el set de entrenamiento aumentado, validación y pruebas.

color a escala de grises con la función `cv2.COLOR_BGR2GRAY`, lo que le permite a la red aprender o concentrarse en formas y texturas, por sobre variaciones de color. Estas aumentaciones permiten obtener 552 ejemplos para el set de entrenamiento.

Se realizaron pruebas de aumentación con desenfoque gaussiano, pero le generaba demasiada incertidumbre a la red. Por otro parte, el desenfoque no es el escenario de una interfaz, como podría ser para modelos que necesiten evaluar objetos en movimiento o en el que la degradación de la imagen puede presentarse como situación, a partir, por ejemplo, de una cámara de baja fidelidad (Dodge & Karam, 2016).

Las capturas de pantalla tenían una dimensión inicial de 1080 x 2400 píxeles, finalizado el proceso de aumentación se redimensionan las imágenes del *set* de entrenamiento, validación y pruebas a 360 x 800 píxeles, tamaño estándar de una interfaz de usuario en dispositivos móviles con sistema *Android*, propuesto en *Figma* para la composición de pantallas en su presentación *Android Large* y también, así, disminuir la carga de trabajo del ordenador durante el entrenamiento.

En la Tabla 1 se presentan ejemplos de las diferentes clases o patrones presentes en el *dataset* y la cantidad de imágenes en el set de entrenamiento, posterior al proceso de aumentación de datos y la cantidad de datos presentes en el *set* de validación y pruebas.

En la Figura 9 se presentan ejemplos de los primeros 12 patrones del *dataset*. En la Figura 10 se presentan ejemplos de los siguientes 12 patrones del *dataset*.

3.2. Modelo Deep learning

LeCun et al., (2016) señalan que el aprendizaje profundo, es una rama de la inteligencia artificial que ha demostrado gran capacidad en el análisis de la estructura de datos, utilizándose para datos gubernamentales, negocios e imágenes. Otra aproximación, también, es planteada por Wang et al., (2019) en que señalan como un método, por parte de una máquina, de superarse y obtener el aprendizaje de nuevas habilidades.

En el caso del modelo propuesto, es un modelo de *Deep learning*¹⁴, basado en redes convolucionales y aprendizaje supervisado, es decir, se han etiquetado los datos en él, para que el modelo sea capaz de identificar las características que poseen las categorías o patrones. Shende et al., (2019) proponen que este proceso lo realizará el modelo desde su entrada a su salida, convertida en una probabilidad o predicción. En la Figura 11 se presenta, a modo de ejemplo, una red convolucional y sus diferentes capas convolucionales, desde la entrada de la imagen, a la salida, entrenada con dos *GPU*¹⁵, por eso se observan dos procesos paralelos.

Deng (2014), también propone tres caracterizaciones del *Deep learning* y sus descripciones:

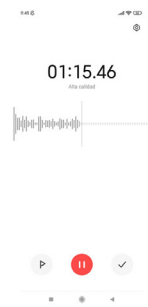
- Redes profundas sin supervisión para aprendizaje generativo.
- Redes profundas para aprendizaje supervisado.
- Redes híbridas.

Para este caso de estudio, la más relevante, es la red profunda para aprendizaje supervisado. Condicionando la capacidad de clasificación a la posibilidad de contar con el etiquetado de los datos, que debe estar disponible de manera directa o indirecta. Es decir, en un *dataset* y en sus anotaciones u organización.

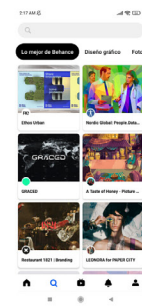
14. Aprendizaje profundo. Denominado así por la cantidad de capas.

15. Graphic processing unit. Placa gráfica.

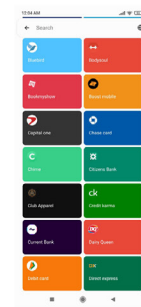
audio_recorder



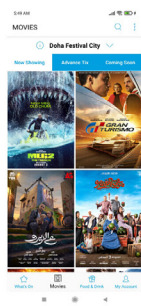
card_grid_md



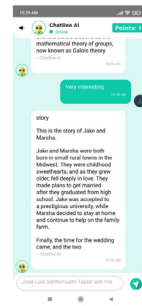
card_grid_sm



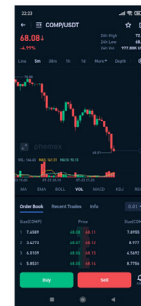
card_grid_xl



conversational



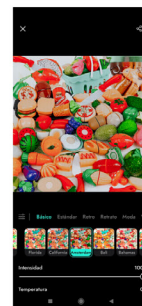
crypto



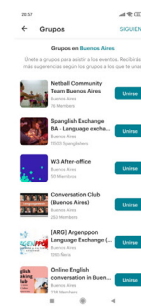
date_range



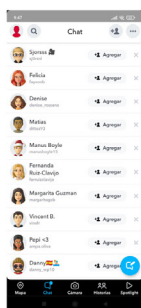
image_filter



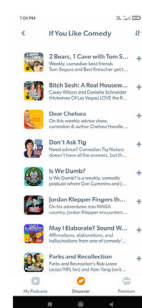
list_md



list_profile



list_sm



list_xl

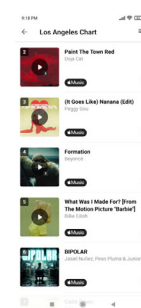
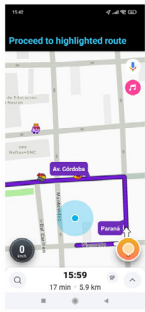
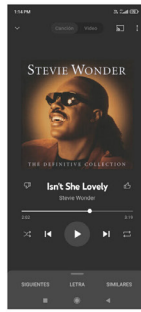


Figura 9. Ejemplos de los 12 primeros patrones o clases presentes en el dataset.

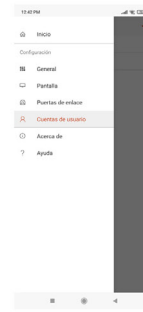
map



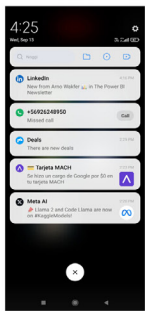
music



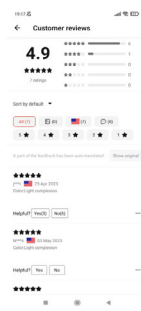
nav_drawer



notification



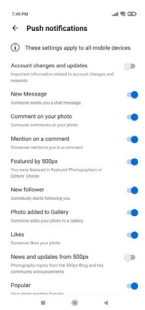
rate



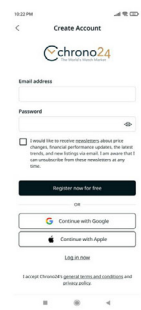
reel



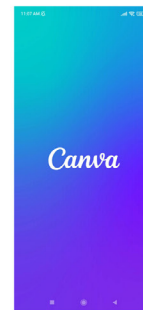
setting



sign



splashscreen



video_fullscreen



walkthrough



weather



Figura 10. Ejemplos de los siguientes 12 patrones o clases presentes en el dataset.

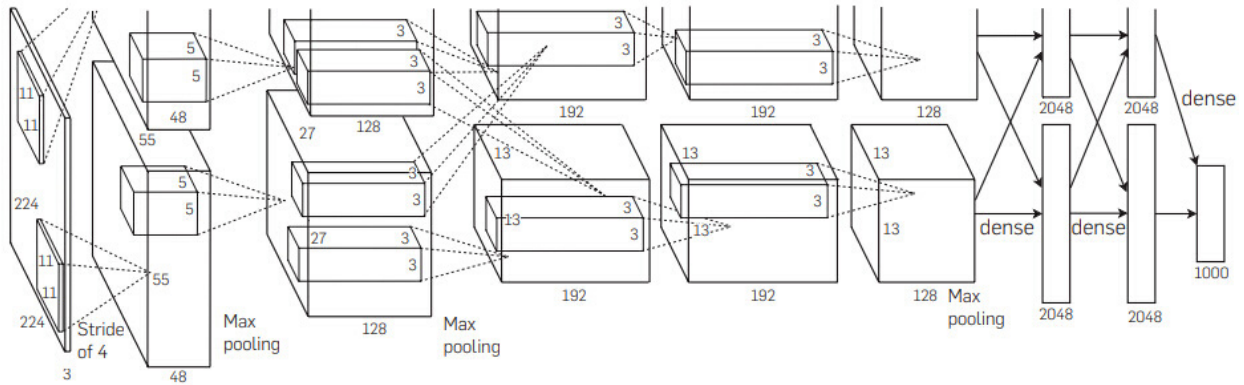


Figura 11.
Red Alexnet, (Krizhevsky, Sutskever & Hinton, 2012).

3.3. Redes convolucionales, EfficientNET

Las redes convolucionales han sido utilizadas ampliamente para problemas de clasificación de imágenes y vídeo (Sharma et al., 2018). Especialmente utilizadas en *el Deep learning*, permiten, a partir de un proceso convolucional, reducir la dimensionalidad de una imagen, capa tras capa, para así obtener un valor de vector final, con el que se contrasta la imagen de entrada sobre la que se necesita realizar una inferencia.

Para el entrenamiento se utiliza *EfficientNET B2*, red convolucional, una red muy eficiente con pequeños conjuntos de datos y recursos informáticos limitados, en su versión *B7* es 8.4x más pequeña y 6.1x más rápida que otras redes convolucionales (Tan & Le, 2020).

La Figura 12 ilustra la comparativa de *EfficientNet* frente a otras redes convolucionales. La red *EfficientNet B2*, obtiene una precisión superior al 80% en el *dataset ImageNet* (Stanford University & Princeton University, 2009). Esta red dispone de pesos pre entrenados que son aprovechados mediante la técnica de *transfer learning*¹⁶.

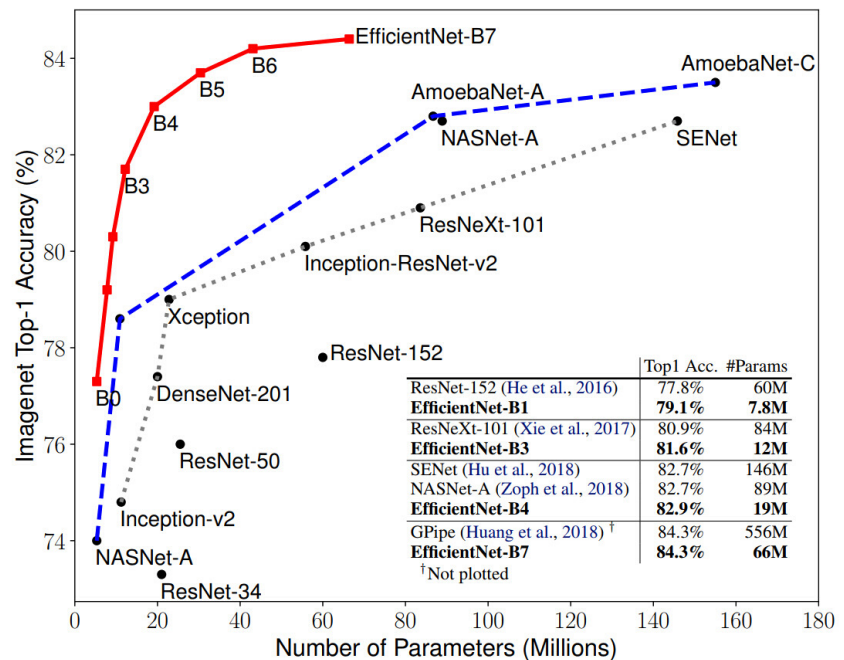


Figura 12.
Comparativa de precisión EfficientNet vs otras redes convolucionales (Tang & Le, 2020).

16. Transferencia de aprendizaje, se aprovecha un aprendizaje anterior en un problema similar.

3.4. Entrenamiento del modelo

Para el entrenamiento se utiliza la red convolucional *EfficientNET*, implementada en el trabajo de Daniel Bourke (Bourke, 2023), utilizando la versión *B2* de la red. Las versiones *B0* y *B1*, al tener menos parámetros, no lograban un buen desempeño, por otra parte, las versiones superiores de la red como *B3*, *B4*, al tener demasiados parámetros, tampoco presentaban una precisión destacable.

Se modificó el trabajo de Bourke (2023), incorporando un *dataloader*¹⁷ para validación y métricas como *F1 score*¹⁸. Se agregó una matriz de confusión y así observar la coincidencia entre clase o ejemplo real y su predicción en la matriz de falsos positivos, falsos negativos. Con lo anterior se busca visualizar, también, qué clases podrían ser ambiguas para el modelo o generar confusión en su clasificación. Se agregó, así mismo, la capacidad de guardar el optimizador en los pesos para futuras inferencias. Se utilizó la documentación de *Pytorch*, de la que se modificó el optimizador de *SGDA* a *ADAM*, esto último a partir de diferentes artículos que lo señalan como un buen optimizador para tareas de clasificación de imágenes (Duman & Süzen, 2022). *ADAM*, además, consume menos recursos y la velocidad de convergencia es más rápida.

La red toma las imágenes del *dataset* y genera lotes aleatorios de 128 imágenes por época, se eliminaron las transformaciones de recorte originalmente propuesta por Bourke (2023), en este problema no tiene sentido obtener el valor de una sección o zona en particular de la interfaz. Se aplicaron durante el entrenamiento solo transformaciones de tamaño, para hacer compatible el tamaño original de las imágenes 360 x 800 píxeles a un tamaño de 260x260 píxeles, que necesita la red como dimensiones de entrada.

Se utiliza el modelo pre-entrenado, lo que supone aprovechar algunas características aprendidas por el modelo en otro problema similar. El concepto de *transfer learning* podría definirse como tomar las características aprendidas en un problema y utilizarlas en un problema nuevo o similar. En otra definición, Según Öztürk et al., (2023) aparece como "un método que acelera el entrenamiento transfiriendo características fuertes de modelos entrenados en una gran cantidad de imágenes". Proponen, también, el *congelamiento*¹⁹ de las capas superiores.

Para este modelo que se presenta, se modificó o congeló la salida, de 1.000 características, que corresponden a las características de *ImageNet*, a 24 características, que son las categorías que se desean clasificar y evaluar. Se utiliza entropía cruzada.

Se implementa el optimizador *ADAM*, con una tasa de aprendizaje de 0.001, y un *dropout* de 0.20, apagando el 20% de las neuronas, para evitar el sobreajuste. El *dropout* es una técnica planteada como útil para evitar el sobreajuste por Srivastava et al., (2014). También, se implementó una regularización *label smooth* de 0.011, para castigar el exceso de confianza, se plantea como técnica de regularización en el artículo *Smooth or Not? When Label Smoothing Meets Noisy Labels* (Wei et al., 2020).

El entrenamiento se realizó por 100 épocas, más allá de esta cantidad de épocas el modelo no experimenta mejoras en sus métricas de precisión y pérdida. Al finalizar el entrenamiento se guardan los pesos en formato *.pt*²⁰.

17. Cargador de datos.

18. Métrica que combina precisión del modelo con recall.

19. No utilizar las capas anteriores de la red.

20. Pesos. Características que aprendió el modelo y con los que se pueden realizar posteriormente las inferencias, sin la necesidad de ejecutar un nuevo entrenamiento.

3.5. Inferencias del modelo. Gradio

Gradio es un módulo de Python que permite integrar y construir, una interfaz gráfica para la inferencia de modelos de *Machine learning* y *Deep learning* (Abid et al., 2019).

Permite cargar los pesos generados en el entrenamiento para realizar inferencias sobre imágenes, contando con una interfaz gráfica de usuario que facilita su uso.

Puede integrarse en un servidor web Python que permite subir o arrastrar imágenes para su posterior inferencia, despliega una pantalla con los resultados, con el nombre de las categorías inferidas y el porcentaje de confianza que tiene en la inferencia. Se pueden personalizar textos y estilos.

En la Figura 13 se presenta la interfaz de *Gradio*, implementada de manera local.

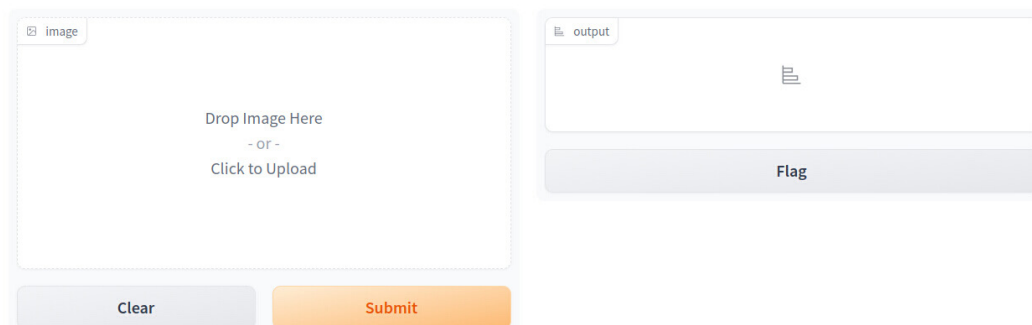


Figura 13.
Interfaz de usuario
Gradio.

4. Resultados y discusión

4.1. Entrenamiento

Los resultados al finalizar las 100 épocas de entrenamiento son bastante alentadores, los porcentajes, tanto de pérdida, como precisión, en los diferentes sets o conjuntos de datos, se aprecian óptimos, no observándose sobreajuste o *underfitting*. También, como una manera de obtener métricas más concluyentes, se integró la métrica *F1 Score*, que es una combinación de precisión y *recall*. En la Tabla 2 se ofrecen los resultados del entrenamiento, en los diferentes sets y la métrica *F1 Score* al finalizar 100 épocas de entrenamiento.

Set	Pérdida/Loss	Precisión/Accuracy	F1 Score
Entrenamiento	0.9287 %	95.63 %	
Validación	0.9379 %	93.59 %	
Prueba	0.9566 %	92.45 %	
			0.9236 %

Tabla 2.
Resultados en set de
entrenamiento, validación y
prueba. Incluye métrica F1
score.

En la Figura 14 se presentan las curvas de precisión del entrenamiento, validación y prueba, al finalizar de 100 épocas.

Nota. La curva azul señala el nivel de precisión del modelo en el set de entrenamiento, en general, siempre es más alta en este *set*, porque son datos que ha visto el modelo. La curva naranja es la de validación, se espera que sea más baja que el entrenamiento, pero estable, se aprecia un buen resultado, generalmente permite evidenciar sobreajuste, en caso de que empeore a lo largo del entrenamiento, mientras la precisión en el set de entrenamiento sigue mejorando. Por último, la curva verde señala la precisión

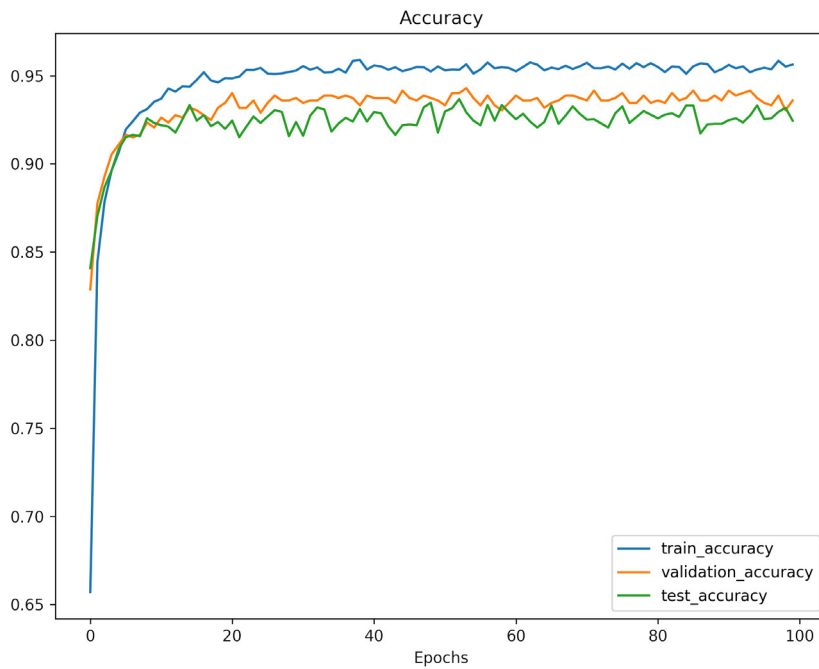


Figura 14. Visualización de las curvas de precisión, en los sets de entrenamiento, validación y pruebas, al finalizar 100 épocas.

del modelo en el set de pruebas, siempre es más baja que las dos curvas anteriores, dado que son datos que no ha visto el modelo durante el set de entrenamiento o validación. Las curvas aparecen como óptimas, tienen buen tiempo de convergencia, prácticamente en las 20 primeras épocas de entrenamiento. Se estabiliza la precisión en las siguientes épocas de entrenamiento. No evidencian sobreajuste o *underfitting*.

En la Figura 15 se presentan las curvas de pérdida del entrenamiento, validación y prueba, al finalizar las 100 épocas.

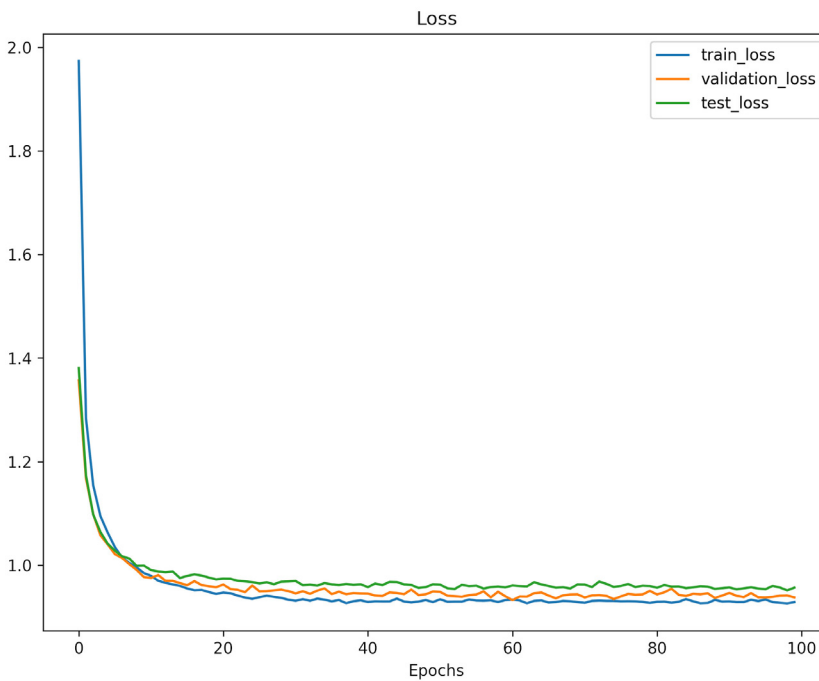


Figura 15. Visualización de las curvas de pérdida, en los sets de entrenamiento, validación y prueba, al finalizar 100 épocas.

Nota. La curva verde señala la pérdida del modelo en el set de pruebas, siempre es más alta que la pérdida en los otros dos sets, dado que son datos que no ha visto el modelo durante el set de entrenamiento o validación. La curva naranja es la curva de validación, se espera que esté al centro, porque es un set diferente al de entrenamiento, pero que permite al modelo ajustar los hiperparámetros. La curva azul señala la del modelo en el set de entrenamiento. Siempre aparece con el nivel más bajo, en todos los procesos de entrenamiento, porque está basado en los datos en que ha sido entrenado el modelo, los conoce. Las curvas aparecen como óptimas, esperables, se reduce la pérdida y se mantiene estable al llegar a las 20 épocas de entrenamiento.

En la Figura 16 se presenta la matriz de confusión, para determinar los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Obtiene buen desempeño en la clasificación de patrones.

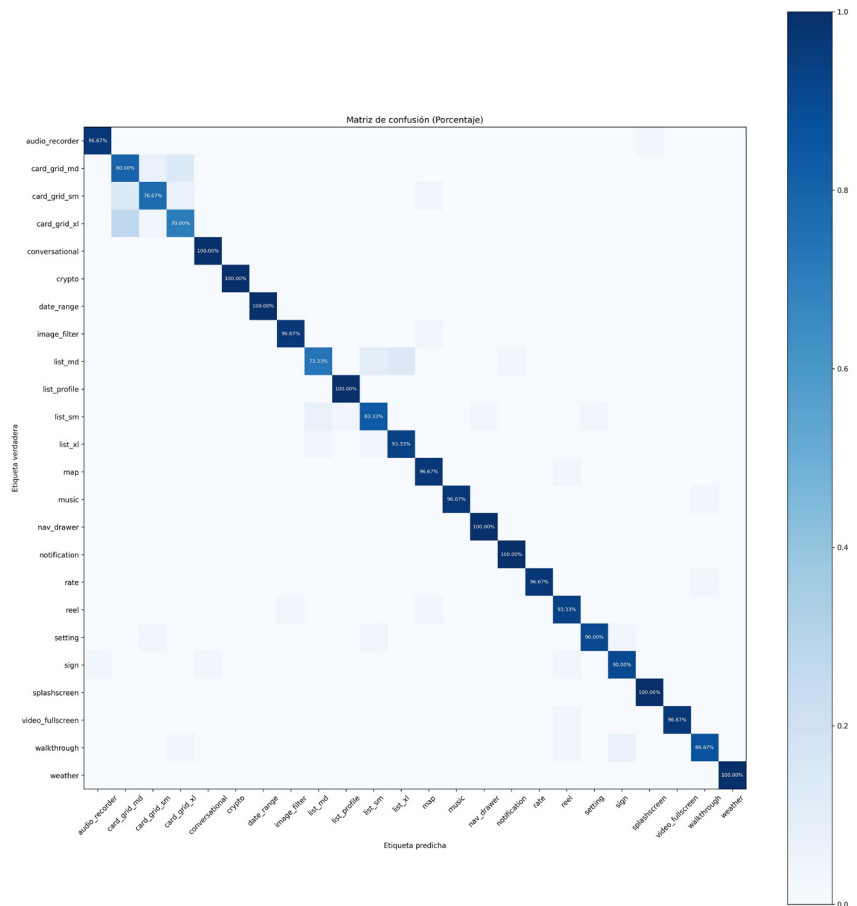


Figura 16. Matriz de confusión

Nota. Matriz para determinar los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos. Permite, además, observar qué patrones generan mayor confusión entre ellos. Se espera una diagonal clara, la opacidad del azul indica alta coincidencia entre la clase real y la inferida. A mayor opacidad, mayor precisión. Con poca opacidad aquellos ejemplos que generan algún grado de confusión. Se observa algo de confusión en las grillas de cards, dado que son bastante parecidas, tienen moderadas variaciones de tamaño y diagramación.

4.2. Inferencia del modelo sobre interfaces

Posterior al entrenamiento se cargan los pesos generados en *Gradio* con sus optimizadores, para facilitar la inferencia o evaluación de interfaces de usuario, arrastrando la imagen al *prompt*. *Gradio* entrega una evaluación de las primeras tres clases sobre las cuales tiene mayor confianza. Ninguna de las imágenes que se le presentan al modelo las ha visto durante el entrenamiento, fueron obtenidas desde *Figma community* (Figma, 2019), en diferentes *drafts*, seleccionando diseños que mantienen consistencia y estándares, que presentan los criterios señalados por *Material Design* y otros que no cumplen con esos criterios, con el objetivo de poner a prueba el modelo.

La acción por parte del usuario para poder realizar la inferencia es solo arrastrar la imagen sobre el *input* de la izquierda, y presionar el botón *submit*, *Gradio* mostrará la inferencia en la sección de la derecha.

4.3. Métricas

Posterior al proceso de entrenamiento se puso a prueba el modelo con interfaces o patrones de diseño no presentes en el *dataset*, con la finalidad de obtener métricas para la interpretación de los resultados. En la Tabla 3 se presentan las métricas, porcentajes de confianza en las inferencias y observaciones para interpretar los resultados. Conviene a los diseñadores o equipos de diseño discutir y analizar los resultados y observaciones propuestas, con la finalidad de mejorar los prototipos y/o determinar qué está fallando en caso de obtener una confianza baja frente a sus diseños.

Porcentaje de confianza en la inferencia	Observaciones
0% - 50%	El patrón o interfaz diseñado tiene problemas severos en términos de diagramación, tipografía, colores, formas o tamaños. Tiene nula consistencia frente al patrón de diseño con el que se contrasta.
51% - 70%	El patrón o interfaz diseñado tiene problemas moderados en términos de diagramación, tipografía, colores, formas o tamaños. Tiene poca consistencia frente al patrón de diseño con el que se contrasta.
71% - 80%	El patrón o interfaz diseñado tiene problemas leves en términos de diagramación, tipografía, colores, formas o tamaños. Tiene una moderada consistencia frente al patrón de diseño con el que se contrasta.
81% - 100 %	El patrón o interfaz diseñado tiene un diseño satisfactorio en términos de diagramación, tipografía, colores, formas o tamaños. Tiene alta consistencia frente al patrón de diseño con el que se contrasta.

Tabla 3.
Resultados en set de entrenamiento, validación y prueba. Incluye métrica F1 score.

4.4. Ejemplos de inferencias y análisis de métricas

En la Figura 17 se aprecia la inferencia sobre una interfaz que permite el inicio de sesión. Obtiene alta confianza, está bien diagramada, bien compuesta, con los elementos centrados y bien definidos, en términos de forma, tamaño y color, por ejemplo, botón de acción principal está destacado, amplio, es muy similar al inicio de sesión actual de un aplicativo móvil, es decir, posee consistencia y estándares actuales.

Nota. Se consultó el perfil de usuario @sanghita en la comunidad de Figma para obtener el prototipo a evaluar (Figma Community, s. f.-a).

En la Figura 18 se le presenta al modelo otra interfaz, para inicio de sesión, que obtiene una baja confianza, no está bien diagramada, tiene problemas de composición, se

Figura 17. Interfaz que obtiene alta confianza en la inferencia.

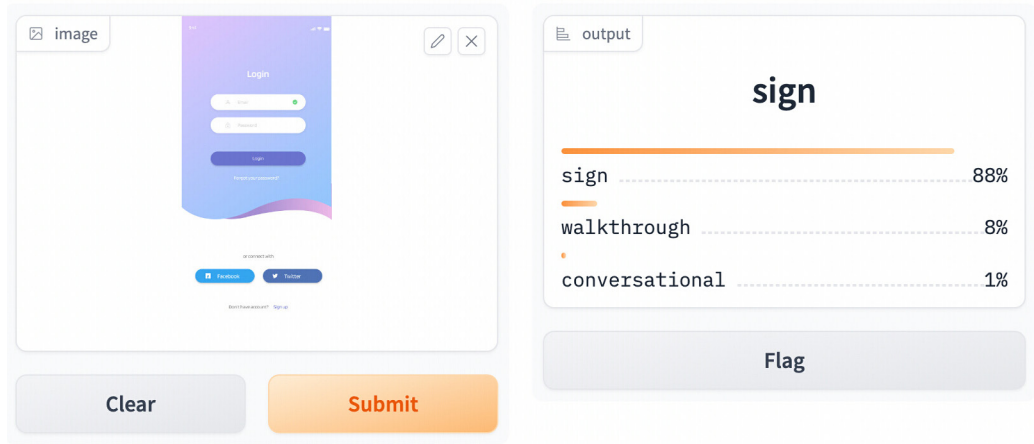


Figura 18. Interfaz que obtiene baja confianza.

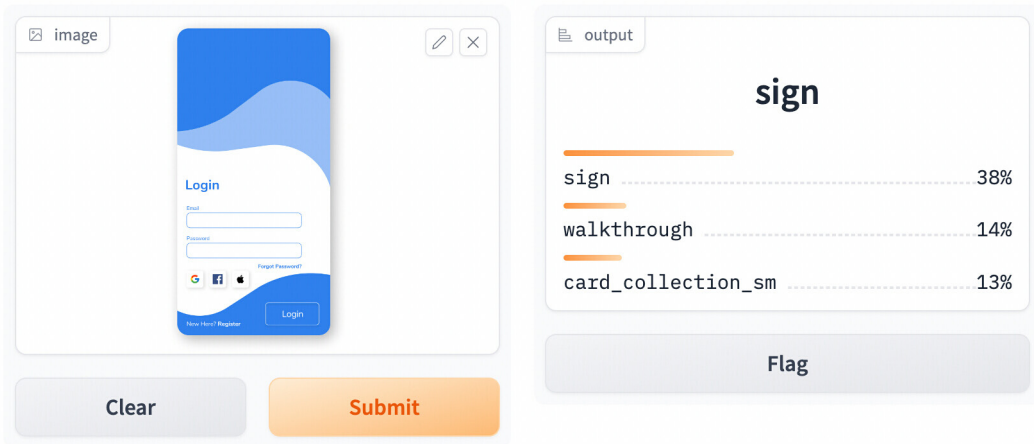
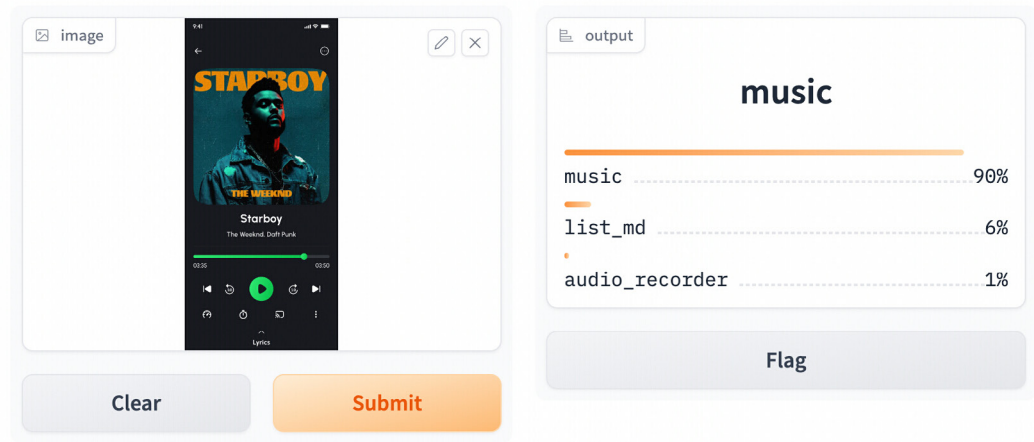


Figura 19. Interfaz que obtiene alta confianza.



encuentra alineada a la izquierda, y el botón *login* mimetizado con parte de la banda azul, pequeño, es el llamado a la acción, faltan elementos como el ícono que permite ver la contraseña, tiene problemas de consistencia y estándares. Además, esas bandas de colores confunden a la red, la hace pensar que es un *Walkthrough* o una pantalla de inducción o tutorial para el primer uso de una aplicación.

Nota. Se utilizó el recurso *Modern Login UI 2.0* de la comunidad de *Figma* como prototipo a evaluar (Figma Community, s. f.-b).

En la Figura 19 se evalúa una interfaz de reproducción de música, obtiene una alta confianza, tiene la mayoría de los elementos de un reproductor: imagen amplia, indicador de progreso e íconos de acciones de un reproductor de música.

Nota. El kit de interfaz de usuario *Hear me—Music Streaming & Podcast App* de la comunidad de *Figma* fue utilizado como prototipo a evaluar (Figma Community, s. f.-c).

En la Figura 20 se le presenta al modelo una interfaz de reproducción de música, que obtiene baja por parte del modelo, tiene poca iconografía de acciones para interactuar con la reproducción y la imagen aplica Gestalt, ley de continuidad, para permitir la reproducción de otras canciones, una funcionalidad que no se incluye en la mayoría de las interfaces de reproducción de música. También, el espacio o aire entre la imagen de la canción y los controles de reproducción es demasiado amplio, considerando que en una interfaz de usuario móvil cada píxel es importante, por el tamaño reducido de la pantalla.

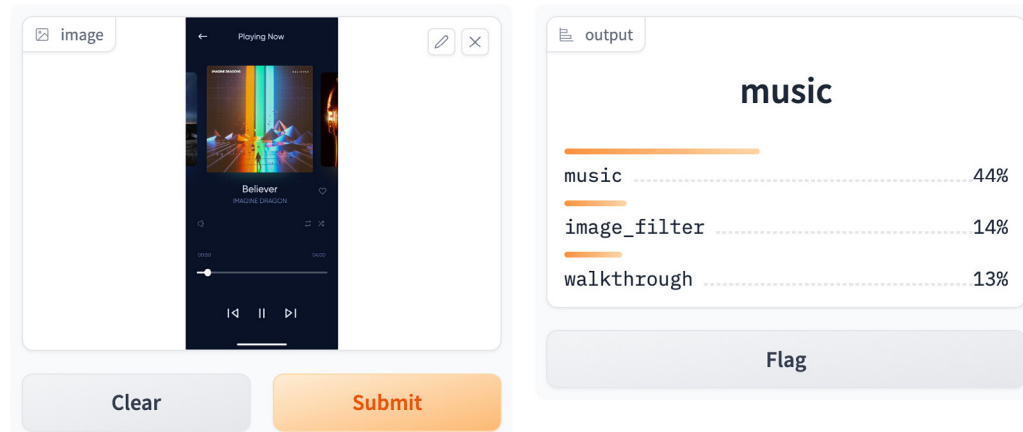


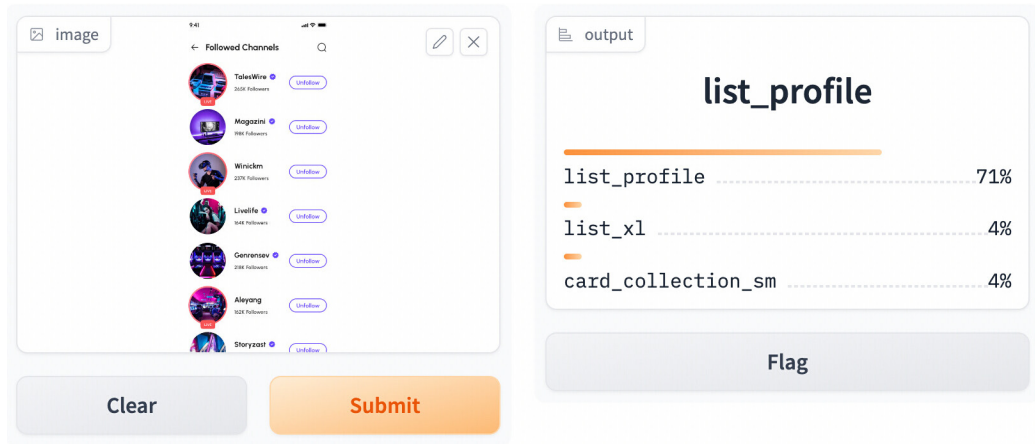
Figura 20
Interfaz que obtiene baja confianza.

Nota. El recurso *Music Player UI KIT (Community)* de la comunidad de *Figma* fue utilizado como prototipo a evaluar (Figma Community, s. f.-d).

En la Figura 21 se aprecia una confianza moderada, el círculo del *avatar* es demasiado grande para este tipo de ítems de la lista, también, posee un rectángulo que destaca a un determinado perfil en la parte inferior del círculo, elementos que no se encuentran en este tipo de componentes, la confianza es moderada.

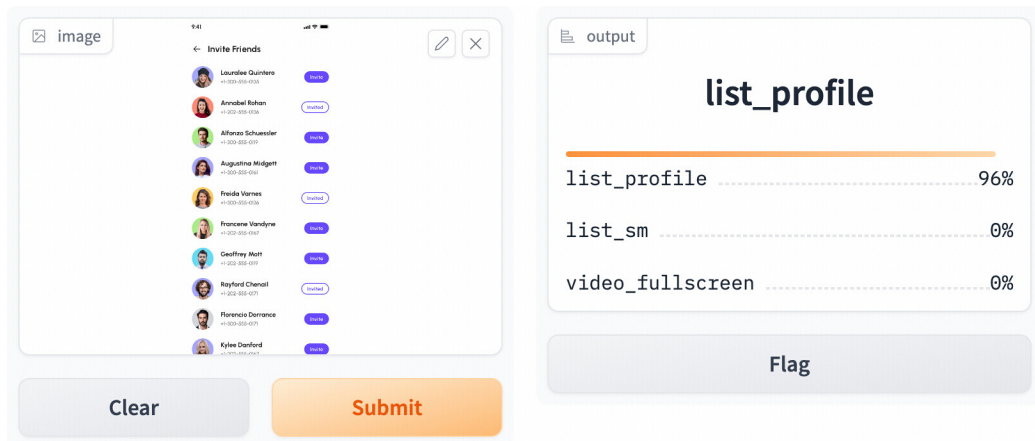
En la Figura 22, se presenta al modelo otra interfaz, una lista de perfiles o seguidores, que obtiene alta confianza, los tamaños de los elementos que conforman los ítems de la lista están más proporcionados, son relativamente estándares a este tipo de componentes e interfaces.

Figura 21
Interfaz que presenta leves problemas de tamaños, diseño con algún grado de inconsistencia.



Nota. Se hizo uso del kit de interfaz de usuario *Streamo—Game Live Streaming App UI Kit* de la comunidad de Figma (Figma Community, s. f.-e).

Figura 22
Interfaz que obtiene alta confianza.



Nota. Se hizo uso del kit de interfaz de usuario *Streamo—Game Live Streaming App UI Kit* de la comunidad de Figma (Figma Community, s. f.-f)

5. Conclusiones

Realizadas las pruebas de laboratorio quedó en evidencia la capacidad del modelo de evaluar prototipos de interfaces de usuario en dispositivos móviles, en términos de confianza y calidad, a partir de la aplicación de los criterios propuestos por *Material Design 3* que se utilizaron en la selección de ejemplos presentes en el *dataset*. De momento la evaluación solo es cuantitativa, pero se espera que, en trabajos posteriores, sea cuantitativa y cualitativa.

Las métricas propuestas pueden ser discutidas, el porcentaje de inferencia y su graduación es una aproximación a la interpretación de los resultados, son todavía inferidos con una matriz y por observación.

En adelante, se espera contar con 350 a 400 ejemplos únicos y ampliar la cantidad de patrones que el modelo es capaz de evaluar.

La integración de *Gradio* y la carga de pesos pre entrenados para realizar las inferencias no demanda demasiados recursos informáticos, permitiendo proyectar su uso sin contar con un gran despliegue de *hardware*.

Como trabajo a futuro, a corto plazo, se espera tener disponible el sistema y modelo en una plataforma web de libre acceso, que permita su uso por parte de estudiantes de diseño o profesionales que requieran evaluar propuestas de diseño de interfaz de usuario o sus actuales aplicaciones en producción, arrastrando las imágenes de sus prototipos a un *prompt* para su evaluación, de esta manera, facilitar la interacción con el sistema y evitar integraciones de ambientes informáticos.

Posterior al proceso de evaluación, en caso de no obtener una evaluación satisfactoria, se desplegarán los mejores ejemplos presentes en el *dataset* y una guía de especificaciones técnicas, que haga énfasis en tipografías, colores, formas, tamaños, elevaciones, proporción y diagramación.

Así mismo, a mediano plazo, se espera introducir la detección de elementos y componentes mediante marcos de trabajo como *YOLO* o *SAM*, como una manera de facilitar el análisis atómico de una interfaz, desagregando los elementos y evaluando su diseño, su calidad. Para esto último se construirá un segundo *dataset*, de al menos 25 clases o categorías, con la finalidad de reforzar el análisis de los parámetros mencionados en el párrafo anterior, aplicado a los elementos de una interfaz de usuario en dispositivos móviles.

También, a mediano plazo, se espera liberar el *dataset* y los pesos pre entrenados del modelo a la comunidad de investigadores para su utilización, para este fin, se entrenará el modelo en diferentes redes, que permitan la utilización de los pesos generados, a modo de pesos pre entrenados, para abordar problemas similares.

Referencias

- Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). *Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild* (Número arXiv:1906.02569). arXiv. <https://doi.org/10.48550/arXiv.1906.02569>
- Adobe. (2023,-a). *Relleno generativo: Adobe Photoshop*. Recuperado el 15 de septiembre de 2023, de <https://www.adobe.com/es/products/photoshop/generative-fill.html>
- Adobe. (2023,-b). *Volver a colorear la ilustración*. Recuperado 28 de noviembre de 2023, de <https://helpx.adobe.com/content/help/cl/es/illustrator/using/re-color-artwork.html>
- Adobe. (2023,-c). *Automated Video Editing with AI*. Recuperado 28 de noviembre de 2023, de <https://www.adobe.com/products/premiere/ai-video-editing.html>
- Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1(1), 100004. <https://doi.org/10.1016/j.ijm.2020.100004>
- Altinbas, M. D., & Serif, T. (2022). GUI Element Detection from Mobile UI Images Using YOLOv5. *Mobile Web and Intelligent Information Systems: 18th International Conference, MobiWIS 2022, Rome, Italy, August 22–24, 2022, Proceedings*, 32-45. https://doi.org/10.1007/978-3-031-14391-5_3
- Bourke, D. (2023). *Learn PyTorch for Deep Learning [Software]*. https://github.com/mrdbourke/pytorch-deep-learning/blob/78be136b27f7ee6a95875f4d-5869339cbc7b89cf/06_pytorch_transfer_learning.ipynb

- Canonical. (2022). *Ubuntu 22.04.3 LTS (Jammy Jellyfish)*. Recuperado el 20 de septiembre de 2023, de <https://releases.ubuntu.com/jammy/>
- Canva. (s. f.). Free AI Image Generator: Online Text to Image App. Recuperado 28 de noviembre de 2023, de <https://www.canva.com/ai-image-generator/>
- Construct. (s. f.). *Game Making Software—Construct 3*. Recuperado 29 de noviembre de 2023, de <https://www.construct.net>
- Chen, J., Chen, C., Xing, Z., Xia, X., Zhu, L., Grundy, J., & Wang, J. (2020). *Wireframe-Based UI Design Search Through Image Autoencoder*. *ACM Transactions on Software Engineering and Methodology*, 29(3), 1-31. <https://doi.org/10.1145/3391613>
- Deka, B., Huang, Z., & Kumar, R. (2016). ERICA: Interaction Mining Mobile Apps. *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 767-776. <https://doi.org/10.1145/2984511.2984581>
- Deka, B., Huang, Z., Franzen, C., Hibsichman, J., Afergan, D., Li, Y., Nichols, J., & Kumar, R. (2017). Rico: A Mobile App for Building Data-Driven Design Applications. *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 845-854. <https://doi.org/10.1145/3126594.3126651>
- Deng, L. (2014). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3-4), 197-387. <https://doi.org/10.1561/20000000039>
- Dodge, S., & Karam, L. (2016). *Understanding How Image Quality Affects Deep Neural Networks* (arXiv:1604.04004). arXiv. <http://arxiv.org/abs/1604.04004>
- Duman, B., & Süzen, A. A. (2022). A Study on Deep Learning Based Classification of Flower Images. *International Journal of Advanced Networking and Applications*, 14(2), 5385-5389. <https://www.proquest.com/docview/2734719954/abstract/55C16F85DA1545E1PQ/1>
- Figma Community. (2019). Figma. Recuperado el 10 de julio de 2023, de <https://www.figma.com/community>
- Figma Community. (2022). Figma. Recuperado el 15 de octubre de 2023, de <https://www.figma.com/community/plugin/746656816456492537/avatars-generator>
- Figma (@sanghita) | Figma Community. (s. f.-a). Figma. Recuperado 17 de septiembre de 2023, de <https://www.figma.com/@sanghita>
- Modern Login UI 2.0 | Figma Community. (s. f.-b). Figma. Recuperado el 17 de septiembre de 2023, de <https://www.figma.com/community/file/888461935200285665/Modern-Login-UI--2.0>
- Hearme—Music Streaming & Podcast App UI Kit | Figma Community. (s. f.-c). Figma. Recuperado el 10 de julio de 2023, de <https://www.figma.com/community/file/1216774653609903227/Hearme---Music-Streaming-&-Podcast-App-UI-Kit>
- Music Player UI KIT (Community) | Figma Community. (s. f.-d). Figma. Recuperado 17 de septiembre de 2023, de [https://www.figma.com/community/file/1093095482562421703/Music-Player-UI-KIT-\(Community\)](https://www.figma.com/community/file/1093095482562421703/Music-Player-UI-KIT-(Community))
- Streamo—Game Live Streaming App UI Kit | Figma Community. (s. f.-e). Figma. Recuperado el 17 de septiembre de 2023, de <https://www.figma.com/community/file/1218093767084140525/Streamo---Game-Live-Streaming-App-UI-Kit>

- Streamo—Game Live Streaming App UI Kit | Figma Community. (s. f.-f). Figma. Recuperado el 17 de septiembre de 2023, de <https://www.figma.com/community/file/1218093767084140525/Streamo---Game-Live-Streaming-App-UI-Kit>
- Atomic Design | Atomic Design by Brad Frost. (2016). Recuperado 10 de mayo de 2023, de <http://atomicdesign.bradfrost.com/table-of-contents/>
- Garrett, J. J. (2011). *The elements of user experience: User-centered Design for the Web and Beyond* (2nd ed., pp. 18-30) New Riders Pub.
- Google Cloud. (s. f.). *Google Cloud AutoML: Entrena modelos sin experiencia en AA*. Recuperado el 29 de noviembre de 2023, de <https://cloud.google.com/automl?hl=es-419>
- Hsiao, Y.-C., Zubach, F., Wang, M., Jindong, & Chen. (2022). *ScreenQA: Large-Scale Question-Answer Pairs over Mobile App Screenshots* (Número arXiv:2209.08199). arXiv. <https://doi.org/10.48550/arXiv.2209.08199>
- IBM (s. f.). *Carbon Design System*. Recuperado 28 de noviembre de 2023, de <https://www.carbondesignsystem.com>
- Kalbach, J. (2016). *Mapping experiences: A Complete Guide to Creating Value Through Journeys, Blueprints, and Diagrams*. O'Reilly Media.
- Keselj, A., Milicevic, M., Zubrinic, K., & Car, Z. (2022). The Application of Deep Learning for the Evaluation of User Interfaces. *Sensors*, 22(23), 9336. <https://doi.org/10.3390/s22239336>
- Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., & Girshick, R. (2023). *Segment Anything* (arXiv:2304.02643). arXiv. <https://doi.org/10.48550/arXiv.2304.02643>
- Kholmatova, A. (2017). *Design systems* (1st ed., pp. 20-28). Smashing Books.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. En F. Pereira, C. J. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems* (Vol. 25). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), Article 7553. <https://doi.org/10.1038/nature14539>
- Leiva, L. A., Hota, A., & Oulasvirta, A. (2020). Enrico: A Dataset for Topic Modeling of Mobile UI Designs. *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*, 1-4. <https://doi.org/10.1145/3406324.3410710>
- Lemley, J., Bazrafkan, S., & Corcoran, P. (2017). Smart Augmentation—Learning an Optimal Data Augmentation Strategy. arXiv.org. <https://doi.org/10.1109/AC-CESS.2017.2696121>
- Li, T. J.-J., Popowski, L., Mitchell, T. M., & Myers, B. A. (2021). Screen2Vec: Semantic Embedding of GUI Screens and GUI Components. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1-15. <https://doi.org/10.1145/3411764.3445049>

- Material Design 3. (2021a). *Material Design*. Recuperado el 7 de septiembre de 2023, de <https://m3.material.io/>
- Material Design 3. (2021b). *Typography: Type scale tokens*. Recuperado el 27 de noviembre de 2023, de <https://m3.material.io/styles/typography/type-scale-tokens>
- Material Design 3. (2021c). *Color system: How the system works*. Recuperado el 27 de noviembre de 2023, de <https://m3.material.io/styles/color/system/how-the-system-works>
- Material Design 3. (2021d). *Lists – Material Design 3. Specs. Anatomy*. Recuperado el 27 de noviembre de 2023, de <https://m3.material.io/components/lists/specs>
- Material Design 3. (2021e). *Lists – Material Design 3. Specs. Measurements*. Recuperado el 27 de noviembre de 2023, de <https://m3.material.io/components/lists/specs>
- Material Design 3. (2021f). *Lists – Guidelines*. Recuperado el 27 de noviembre de 2023, de <https://m3.material.io/components/lists/guidelines>
- Microsoft. (s. f.). *Fluent UI - Get started—Fluent UI*. Recuperado 28 de noviembre de 2023, de <https://developer.microsoft.com/en-us/fluentui#/>
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (Número arXiv:1301.3781). arXiv. <http://arxiv.org/abs/1301.3781>
- Moghadam, M. M., Boroomand, B., Jalali, M., Zareian, A., Daeijavad, A., Manshaei, M. H., & Krunz, M. (2022). *Game of GANs: Game-Theoretical Models for Generative Adversarial Networks* (arXiv:2106.06976). arXiv. <http://arxiv.org/abs/2106.06976>
- Nelson, G. S. (2019). Bias in Artificial Intelligence. *North Carolina Medical Journal*, 80(4), 220-222. <https://doi.org/10.18043/ncm.80.4.220>
- Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 152-158. <https://doi.org/10.1145/191666.191729>
- Open CV (s. f.). *OpenCV*. Recuperado 10 de mayo de 2023, de <https://opencv.org/>
- Project Jupyter. (2014). Recuperado el 20 de septiembre de 2023, de <https://jupyter.org>
- PyTorch 2.0 | PyTorch. (2023). Recuperado el 20 de septiembre de 2023, de <https://pytorch.org/get-started/pytorch-2.0/>
- Open AI (2022a, Octubre 25). *DALL·E now Available Without Waitlist*. Recuperado el 10 de marzo de 2023, de <https://openai.com/blog/dall-e-now-available-without-waitlist>
- Open AI (2022b, Noviembre 30). *Introducing ChatGPT*. Recuperado el 10 de marzo de 2023 de <https://openai.com/blog/chatgpt>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You Only Look Once: Unified, Real-Time Object Detection* (Número arXiv:1506.02640). arXiv. <https://doi.org/10.48550/arXiv.1506.02640>

- Santorcuato-Tapia, J. L., & Lloret, J. (2023, julio). *Evaluación del diseño de interfaces de usuario con inteligencia artificial*. Ponencia presentada en el XIV Congreso Latinoamericano de Enseñanza del Diseño, Universidad de Palermo, Buenos Aires, Argentina.
- Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Computer Science*, 132, 377-384. <https://doi.org/10.1016/j.procs.2018.05.198>
- Shahinfar, S., Meek, P., & Falzon, G. (2020). *How many images do I need? Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring* (Número arXiv:2010.08186). arXiv. <http://arxiv.org/abs/2010.08186>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929-1958.
- Srivastava, Y., Murali, V., Dubey, S. R., & Mukherjee, S. (2020). *Visual Question Answering using Deep Learning: A Survey and Performance Analysis* (Número arXiv:1909.01860). arXiv. <https://doi.org/10.48550/arXiv.1909.01860>
- Stability AI (2022, Agosto 22). *Stable Diffusion Online*. *Stable Diffusion web*. Recuperado el 24 de marzo de 2023, de <https://stablediffusionweb.com/>
- Stanford University & Princeton University (2009). *Imagenet*. Recuperado el 10 de marzo de 2023, de <https://www.image-net.org/index.php>
- Tan, M., & Le, Q. V. (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* (Número arXiv:1905.11946). arXiv. <http://arxiv.org/abs/1905.11946>
- Tidwell, J., Brewer, C., & Valencia-Brooks, A. (2019). *Designing Interfaces, 3rd edition*. O'Reilly Media.
- Uber. (2023). *Base Design System*. Recuperado 28 de noviembre de 2023, de <https://base.uber.com/6d2425e9f/p/294ab4-base-design-system>
- UIED. (2020). *UIED online*. Recuperado 17 de diciembre de 2023, de <http://uied.online/>
- Ultralytics. (2020). *YOLOv5*. Recuperado 18 de diciembre de 2023, de <https://docs.ultralytics.com/models/yolov5>
- Vesselov, S., & Davis, T. (2019). *Building design systems: Unify User Experiences through a Shared Design Language*. Apress.
- Wang, H., Ma, C., & Zhou, L. (2009). A Brief Review of Machine Learning and Its Application. *2009 International Conference on Information Engineering and Computer Science*, 1-4. <https://doi.org/10.1109/ICIECS.2009.5362936946>. arXiv. <http://arxiv.org/abs/1905.11946>
- Wei, J., Liu, H., Liu, T., Niu, G., Sugiyama, M., & Liu, Y. (2022). *To Smooth or Not? When Label Smoothing Meets Noisy Labels* (Número arXiv:2106.04149). arXiv. <http://arxiv.org/abs/2106.04149>

WordPress (s. f.). *WordPress, a tu manera*. Recuperado 29 de noviembre de 2023, de <https://wordpress.com/es/>

Wu, J., Wang, S., Shen, S., Peng, Y.-H., Nichols, J., & Bigham, J. P. (2023). *WebUI: A Dataset for Enhancing Visual UI Understanding with Web Semantics* (Número arXiv:2301.13280). arXiv. <https://doi.org/10.48550/arXiv.2301.13280>

Xie, M., Feng, S., Xing, Z., Chen, J., & Chen, C. (2020). UIED: A hybrid tool for GUI element detection. *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1655-1659. <https://doi.org/10.1145/3368089.3417940>