

Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales
Carrera de Informática



TESIS DE GRADO

**APLICACIÓN CON REALIDAD AUMENTADA Y API
DE INTELIGENCIA ARTIFICIAL PARA DIBUJAR
BOCETOS SENCILLOS CON LÍNEAS**

Tesis de grado para obtener el Título de Licenciatura en Informática
Mención en Ingeniería de Sistemas Informáticos

POR: ALBERTH BERNARTH CONDORI QUISPE

TUTOR: LIC. BRÍGIDA ALEXANDRA CARVAJAL BLANCO

La Paz – Bolivia

2021

HOJA DE CALIFICACIONES
UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA

Tesis de grado:

**APLICACIÓN CON REALIDAD AUMENTADA Y API DE INTELIGENCIA
ARTIFICIAL PARA DIBUJAR BOCETOS SENCILLOS CON LÍNEAS**

Presentado por: Alberth Bernarth Condori Quispe

Para optar el grado Académico de Licenciado en Informática

Mención Ingeniería de Sistemas Informáticos

Nota Numeral: 75

Nota Literal: Setenta y cinco

Ha sido: Acreedor, al Grado Académico de: LICENCIADO EN INFORMÁTICA,
Mención: INGENIERÍA DE SISTEMAS INFORMÁTICOS

Director de la carrera de Informática: Ph.D. Jose María Tapia Baltazar

Tutor: Lic. Brígida Alexandra Carvajal Blanco

Tribunal: P.Ph.D. Cuenca Sarzuri Yohoni

Tribunal: Lic. Tarquino Peralta Celia E.

Tribunal: Lic. Pozo Diaz Victor P.

Presidente Tribunal: M. Sc. Flores Morales Rosa



Universidad Mayor de San Andrés
Facultad de Ciencias Puras y Naturales
Carrera de Informática



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) Visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) Copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) Copiar textualmente parte(s) de su contenido mencionado la fuente y/o haciendo la referencia correspondiente respetando las normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVA EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LEY DERECHOS DE AUTOR.

DEDICATORIA

La presente tesis está dedicada a Dios por haberme permitido realizar mis estudios. A mi madre, quien nunca dejo de apoyarme en mi formación como profesional y persona. A mi hermana por ayudarme a continuar con mis estudios.

Por último, a todos aquellos que tienen ideas que no se han realizado aún.

AGRADECIMIENTOS

Le agradezco a Dios por haberme acompañado a pesar de mis errores siempre ha estado conmigo, representando mi voluntad en los momentos de duda y por brindarme la oportunidad de estudiar.

Le doy gracias a mis padres Juana y Santos por haberme apoyado en muchos aspectos de mi vida no solo en mi formación profesional, así también en mi formación como persona.

alberthbernarth@gmail.com

RESUMEN

Un boceto es una forma de dibujo que nos ayuda a visualizar lo que puede ser un dibujo final. Para realizar bocetos de forma rápida y reproducirlos se emplean técnicas sobre un dibujo que ya está hecho a través de la transposición de un papel blanco sobre otro o el empleo de papel carbón.

El empleo de una aplicación web con tecnologías de realidad aumentada permite a una gran cantidad de usuarios el poder apreciar de esta tecnología en una gran variedad de dispositivos. Así también el uso de una API en una aplicación web permite conectar diversas tecnologías sin que estas dependan de una a la otra, dando la posibilidad de aportar funcionalidades adicionales a la misma.

La presente investigación tiene por objetivo integrar en una aplicación web, con ayuda de un framework, tecnologías de realidad aumentada y API de inteligencia artificial para obtener un boceto rápido y comparar este trabajo con el boceto de referencia y así obtener un resultado visible.

Otro aspecto que se propone es poder integrar varias tecnologías de uso libre en una aplicación web, para facilitar el acceso y la divulgación del proyecto para su continua actualización.

Palabras clave: realidad aumentada, API, aplicación web, inteligencia artificial, boceto, framework, método científico.

ABSTRACT

A sketch is a manner of drawing that helps us visualize what a final drawing can be. To make sketches quickly and reproduce them, techniques are used on a drawing that has already been made through the transposition of one white paper on another or the use of carbon paper.

The use of a web application with augmented reality technologies allows a large number of users to appreciate this technology on a wide variety of devices. Likewise, the use of an API in a web application allows the connection of various technologies without them depending on one another, giving the possibility of providing additional functionalities to it.

The objective of this research is to integrate in a web application, with the help of a framework, augmented reality technologies and artificial intelligence API to obtain a quick sketch and compare this work with the reference sketch and thus obtain a visible result.

Another aspect that is proposed is to be able to integrate various technologies for free use in a web application, to facilitate access and dissemination of the project for continuous updating.

Keywords: augmented reality, API, web application, artificial intelligence, sketch, framework, scientific method.

INDICE

CAPÍTULO 1 MARCO INTRODUCTORIO	1
1.1. INTRODUCCIÓN	1
1.2. ANTECEDENTES	2
1.3. PLANTEAMIENTO DEL PROBLEMA	4
1.3.1. PROBLEMA CENTRAL.....	6
1.3.2. PROBLEMAS SECUNDARIOS	6
1.4. DEFINICIÓN DE OBJETIVOS	6
1.4.1. OBJETIVO GENERAL	6
1.4.2. OBJETIVOS ESPECÍFICOS	7
1.5. HIPÓTESIS.....	7
1.5.1. OPERACIONALIZACIÓN DE VARIABLES	7
1.6. JUSTIFICACIÓN	8
1.6.1. JUSTIFICACIÓN ECONÓMICA	8
1.6.2. JUSTIFICACIÓN SOCIAL	8
1.6.3. JUSTIFICACIÓN CIENTÍFICA	8
1.7. ALCANCES Y LÍMITES	8
1.7.1. ALCANCES.....	8
1.7.2. LÍMITES	9
1.8. METODOLOGÍA	9
CAPÍTULO 2 MARCO TEÓRICO.....	11
2.1. INTRODUCCIÓN	11
2.2. EL DIBUJO	11
2.2.1. EL BOCETO	12
2.2.2. UTILIDAD DEL BOCETO	12
2.2.3. VENTAJAS DEL BOCETO.....	12
2.3. REALIDAD AUMENTADA	13
2.3.1. FUNCIONAMIENTO BÁSICO	13
2.3.2. ELEMENTOS	14
2.3.3. CARACTERÍSTICAS DE LA REALIDAD AUMENTADA.....	15
2.4. REALIDAD AUMENTADA EN LA WEB	16

2.4.1. A-FRAME.....	17
2.4.2. AR.js.....	17
2.4.2.1. TIPOS DE REALIDAD AUMENTADA.....	18
2.4.2.2. REQUERIMIENTOS.....	19
2.4.2.3. DESPLIEGUE.....	20
2.5. APLICACIONES WEB Y SITIOS WEB.....	20
2.5.1. APLICACIÓN WEB.....	21
2.5.2. SITIO WEB.....	21
2.5.3. PRINCIPALES DIFERENCIAS.....	22
2.6. FRAMEWORK.....	22
2.7. DJANGO.....	23
2.7.1. CARACTERÍSTICAS.....	24
2.7.2. FRAMEWORK DOGMÁTICO.....	25
2.7.3. ESTRUCTURA TRADICIONAL DE UN PROYECTO.....	26
2.7.4. PATRÓN DE DISEÑO.....	28
2.8. API.....	29
2.8.1. API REST.....	30
2.9. INTELIGENCIA ARTIFICIAL.....	31
2.9.1. TIPOS DE INTELIGENCIA ARTIFICIAL.....	32
2.9.1.1. INTELIGENCIA ARTIFICIAL ESTRECHA.....	32
2.9.1.2. INTELIGENCIA ARTIFICIAL GENERAL.....	32
2.9.1.3. SUPER INTELIGENCIA ARTIFICIAL.....	32
2.9.2. MEDIDA DEL ÍNDICE DE SIMILITUD ESTRUCTURAL.....	33
2.9.2.1. ENTRADAS Y SALIDAS.....	34
2.9.2.2. ÍNDICE DE SIMILITUD ESTRUCTURAL.....	34
2.10. MÉTODO CIENTÍFICO.....	35
2.10.1. PASOS DEL MÉTODO CIENTÍFICO.....	36
2.11. INGENIERÍA DE SOFTWARE.....	37
2.11.1. SCRUM.....	38
2.11.1.1. CARACTERÍSTICAS.....	38
2.11.1.2. HITOS DE LA METODOLOGÍA DE TRABAJO SCRUM.....	39

2.12. METODOLOGÍA PARA EL DESARROLLO REALIDAD AUMENTADA.....	39
2.12.1. PROBLEMAS DE PROCESO.....	40
2.12.2. HERRAMIENTAS.....	40
CAPÍTULO 3 MARCO APLICATIVO.....	42
3.1. INTRODUCCIÓN	42
3.2. REALIDAD AUMENTADA CON AR.js.....	43
3.3. METODOLOGÍA PROPUESTA	44
3.4. EXPERIMENTACIÓN.....	44
3.4.1. PLANEACIÓN	44
3.4.2. DESARROLLO	48
3.4.2.1. SPRINT 1 CONEXIÓN A SERVICIOS EN LA NUBE	48
3.4.2.2. SPRINT 2 ADMINISTRACIÓN CON FORMULARIOS	56
3.4.2.3. SPRINT 3 MANEJO DE UNA API.....	62
3.4.2.4. SPRINT 4 INTEGRACIÓN DE REALIDAD AUMENTADA.....	67
CAPÍTULO 4 PRUEBA DE HIPÓTESIS.....	73
4.1. INTRODUCCIÓN	73
4.2. ANÁLISIS	73
4.2. EXPERIMENTACIÓN.....	73
4.3. ANÁLISIS	80
CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES	81
5.1. INTRODUCCIÓN	81
5.2. CONCLUSIONES	81
5.3. RECOMENDACIONES.....	82
Bibliografía	83

ÍNDICE DE FIGURAS

Figura 2.1. El boceto frente a un dibujo terminado	12
Figura 2.2. Esquema general del funcionamiento de la realidad aumentada	14
Figura 2.3. Pasos para mostrar una página	27
Figura 2.4. Servicio de un API REST	30
Figura 2.5. Representación de una imagen en un ordenador	34
Figura 2.6. Pasos del método científico.....	37
Figura 2.7. Hitos del proceso Scrum	39
Figura 2.8. Fases para el desarrollo con realidad aumentada	41
Figura 3.1. Un marcador por defecto conteniendo un código QR.....	43
Figura 3.2. Estructura de la aplicación web.....	46
Figura 3.3. Diagrama de casos de uso de la aplicación web	48
Figura 3.4. Diagrama de casos de uso – Desarrollador	49
Figura 3.5. Diagrama de casos de uso – Administrador.....	49
Figura 3.6. Configuración inicial para el alojamiento de archivos.....	54
Figura 3.7. Clave de acceso a una cuenta en Amazon Web Services.....	54
Figura 3.8. Configuración de claves de acceso para Amazon Web Services	55
Figura 3.9. Permisos de acceso a los archivos.....	55
Figura 3.10. Organización de carpetas y objetos en la nube	56
Figura 3.11. Diagrama de casos de uso, creación de formularios – Desarrollador	57
Figura 3.12. Diagrama de casos de uso, creación de formularios – Administrador.....	57
Figura 3.13. Modelo Django para imágenes.....	61
Figura 3.14. Modelo de formularios para imágenes.....	61
Figura 3.15. Imagen a blanco y negro con OpenCV	61
Figura 3.16. Obtener partes representativas de la imagen	62
Figura 3.17. Diagrama de casos de uso, manejo de API – Desarrollador	62
Figura 3.18. Diagrama de casos de uso, manejo de API – Usuario simple.....	63
Figura 3.19. Obtención de resultado por response.	67
Figura 3.20. Obtención de resultado por un método	67
Figura 3.21. Diagrama de casos de uso, integración de realidad aumentada	68

Figura 3.22. Diagrama de caso de uso, interacción con realidad aumentada	68
Figura 3.23. Elementos para mostrar	69
Figura 3.24. Elemento para reconocer	70
Figura 3.25. Implementación de elementos para mostrar	70
Figura 3.26. Reestablecer el tamaño de la imagen	71
Figura 3.27. Muestra de la vista con realidad aumentada	72
Figura 4.1. Boceto número 1	74
Figura 4.2. Boceto número 2	75
Figura 4.3. Boceto número 3	75
Figura 4.4. Resultados con el primer boceto	76
Figura 4.5. Resultados con el segundo boceto.....	77
Figura 4.6. Resultado con el tercer boceto	77
Figura 4.7. Región de aceptación	79

ÍNDICE DE TABLAS

Tabla 2.1. Diferencias entre aplicación y sitio web.....	22
Tabla 3.1. Requerimientos.....	46
Tabla 3.2. Actores que intervienen en la aplicación web	47
Tabla 3.3. Sprint 1 – Servicios en la nube	50
Tabla 3.4. Historias de usuario – Conexión de servicios en la nube	51
Tabla 3.5. Historias de usuario – Funciones para Administrador	52
Tabla 3.6. Prueba de historias de usuario – Servicios en la nube.....	53
Tabla 3.7. Sprint 2 – Administración con formularios	58
Tabla 3.8. Historias de usuario – Creación de formularios	58
Tabla 3.9. Historias de usuario – Manejo de formularios	59
Tabla 3.10. Prueba de historias de usuario. Servicios en la nube.....	60
Tabla 3.11. Sprint 3 – Manejo de API.....	63
Tabla 3.12. Historias de usuario – Conexión con API	64
Tabla 3.13. Historias de usuario – Funciones para Usuario Simple.....	65
Tabla 3.14. Prueba de historias de usuario. Servicios en la nube.....	66
Tabla 4.1. Puntajes tomados de la API.....	78

ÍNDICE DE ECUACIONES

Ecuación 2.1. Índice de similitud estructural	35
Ecuación 4.1. Media muestral	78
Ecuación 4.2. Estadístico de contraste.....	79
Ecuación 4.3. Desviación típica muestral.....	79
Ecuación 4.4. Resultado estadístico de contraste	79

CAPÍTULO 1

MARCO INTRODUCTORIO

1.1. INTRODUCCIÓN

La realidad aumentada es una de las tecnologías de mayor tendencia justo ahora, convirtiéndose cada vez más en un tipo de tecnología que podemos apreciar y experimentar en la mayoría de los teléfonos inteligentes y dispositivos con cámara del mercado. Siendo la realidad aumentada una herramienta que nos permite integrar diferentes entornos justo frente a nosotros con un aumento digital superpuesto. (The Franklin Institute, 2020)

Con tecnologías de realidad aumentada es posible mezclar la realidad tangible con el mundo virtual. En esta se pueden superponer diferentes capas de información que le agregan información a la realidad. Esta tecnología ofrece diversas posibilidades de aplicación en multitud de campos, así como en el mundo del arte tradicional. Logrando así que el arte tradicional presente algún tipo de interacción con el mundo virtual. (Hernández, 2016) Es en este ámbito que surge la aplicabilidad de la realidad aumentada en el arte del dibujo.

Un dibujo puede ser tan simple como trazar simples líneas sobre un papel con algún sentido aparente y también puede ser tan complejo como el diseño de un plano arquitectónico que representa la estructura completa o parcial de alguna construcción o prototipo. De cualquiera de esas dos formas y entre otros usos del dibujo, este es capaz de transmitir una idea o un concepto, por medio de trazos, símbolos, o signos. El dibujo, al igual que el lenguaje, es capaz de expresar o comunicar a través de signos, (Real Academia Española, 2020) donde estos signos son trazos que tienen algún significado. Siendo así el dibujo una forma de comunicar, así como el lenguaje natural o el lenguaje de señas.

Mejorar esta habilidad para realizar dibujos requiere, como muchas otras tareas, de práctica. En el caso del dibujo practicarlos es algo tan sencillo como ponerse a trazar cualquier idea u objeto. La simple práctica del dibujo requiere dibujar algo y repetir, manteniendo, por ejemplo, un cuaderno lleno de dibujos. (Lundgren, 2019)

Con la realidad aumentada y la posibilidad de superponer contenido digital en nuestro entorno real, es posible visualizar o interactuar con contenido artístico en esta, como información sobre el mismo o específicamente con dibujos.

Cuando se realiza una pieza de arte la calificación, observación o apreciación de esta puede ser subjetiva u objetiva. Cuando el dibujo se toma como la representación de trazos existentes se puede calificar de forma objetiva si se toman parámetros que se puedan medir, sin embargo, existen varios parámetros que se pueden tomar en cuenta a la hora de calificar un trabajo artístico como es un dibujo que pretende reproducir otro, como por ejemplo la precisión de los trazos que tiene el dibujo en referencia con el original o el parecido que tiene en relación a un conjunto de otros dibujos. Es así que podemos también, desde esta perspectiva, calificar un dibujo utilizando referencias. (Mojelski, 2017)

Las API de inteligencia artificial de procesamiento de imágenes nos permiten realizar diferentes tareas sobre imágenes individuales o un grupo de ellas, es así que existen APIs que permiten comparar dos imágenes como un conjunto de datos que después de ser procesados retornan un valor que muestra cuán similares visualmente son dos imágenes. (Stonesifer, 2018). Obteniendo así un valor que representa la calificación que un dibujo tiene en referencia a otro.

En el presente trabajo se pretende usar los beneficios de la realidad aumentada para sobreponer contenido digital, que para el contexto presente son bocetos sencillos existentes, y lograr así reproducir o copiar sobre una hoja de papel el contenido digital. Finalmente, con una API disponible de inteligencia artificial medir que tan similar visualmente es el trabajo realizado en la realidad con el contenido digital ofrecido para dibujar.

1.2. ANTECEDENTES

La realidad aumentada es una tecnología que complementa la percepción e interacción con el mundo real y permite al usuario estar en un entorno real aumentado con la información adicional generada por el ordenador. (Basogain, Olabe, Espinosa, Rouéche y Olabe, 2017)

La realidad aumentada, en su dimensión de generar espacios en los que se combina el mundo real con el virtual, ha dado lugar a nuevas perspectivas dentro de la creación artística de vanguardia, con nuevas visiones e interpretaciones. Frente a otros ámbitos del conocimiento, el campo del arte representa uno de los más fructíferos en cuanto al tratamiento de la realidad aumentada, que no sólo se plantea desde un punto de vista tecnológico, sino que además ha realizado nuevas experiencias en las que se juega con la concepción de interconectar dos mundos en los que lo ficticio y lo real conviven en un mismo espacio. (Ruiz, 2014)

Por otra parte, el dibujo no está alejado del uso de las tecnologías, se ha visto la aplicación del arte sobre varias tecnologías actuales en los últimos años. Al ser el dibujo una forma de expresión gráfica sobre un espacio plano, al igual que la escritura se le considera una forma de comunicarse, y por eso también es conocido como un lenguaje universal que ha sido utilizado por la humanidad para transmitir ideas, proyectos y en un sentido más amplio su cultura. (Auquilla, 2013)

Trabajos de investigación

- Innovación educativa basada en realidad aumentada: Una experiencia en el área de dibujo.

El trabajo reconoce la importancia de la integración de las tecnologías de información con la enseñanza del dibujo. Reconoce el amplio desarrollo de tecnologías de imagen, video, video 3D y animación, realidad virtual, realidad aumentada, etc. De esta manera el trabajo pretende implementar la realidad aumentada para la enseñanza del dibujo en el espacio construyendo objetos. (Avilés, Mirete y Maquilón, 2016)

- Aplicaciones de Realidad Virtual y Realidad Aumentada como soporte a la enseñanza del Dibujo Técnico

El objetivo del trabajo fue desarrollar un entorno que le permitiera a un estudiante formar una figura sobre una vista isométrica a través de otras figuras tridimensionales. Para

completar esta actividad el alumno requiere, en su imaginario, girar el objeto tridimensional para visualizar y dibujar cada una de las vistas solicitadas. La aplicación de realidad virtual tiene el fin de asistir y verificar este proceso. (Alvarado, 2019)

- Realidad aumentada en dibujo técnico

El trabajo presenta a la realidad aumentada como una tecnología que mezcla el mundo real con el mundo virtual. Donde en ella se superponen diferentes capas de información a nuestro mundo físico ofreciendo una nueva experiencia de la realidad. Esta experiencia ofrece multitud de posibilidades de aplicación, como la interacción. Hace de la realidad aumentada una herramienta muy interesante en el ámbito educativo. Hoy día contamos con los dispositivos necesarios para su uso. Por eso a lo largo de este proyecto se observa cómo funciona, qué posibilidades ofrece y cómo se puede incluir en la práctica docente, en concreto dentro de la asignatura de Dibujo Técnico. (Hernández, 2016)

Proyectos existentes

- SketchAR TM, ARE YOU READY TO RAISE YOUR CREATIVITY TO THE NEXT LEVEL?

El proyecto es una aplicación móvil, con soporte para Android y IOS, que tiene como propósito hacer fácil adentrarse en el mundo de la creatividad, y obtener habilidades de dibujo a través de visión por computadora, Machine Learning y Redes neuronales, para construir un asistente de dibujo. El uso del proyecto tiene un coste mensual y tiene acceso a toda la plataforma completa. (Drobotko, 2021)

1.3. PLANTEAMIENTO DEL PROBLEMA

El dibujo es una actividad que cualquier persona puede aprender a realizar con práctica, al ser el dibujo un área del arte, sólo se requiere de percepción para realizar dibujos que causen una impresión a la vista y esto sería suficiente para llamar a dicho trabajo una obra de arte desde ciertas perspectivas y criterios subjetivos. Muchos no realizan esta actividad por el pensamiento de que no tienen la habilidad suficiente para dibujar. Por esta razón

algunos toman clases de dibujo o clases de arte, pero al requerir de práctica constante, muchas de estas personas se ven en una situación en la que deciden dejar el curso y llegar a la idea de que dibujar requiere de mucho trabajo o simplemente talento. (Rubio, 2015)

Es evidente que hay determinadas disciplinas en las artes, como el dibujo, en las que el talento innato pareciera ser esencial para poder dedicarse a ello de forma seria. Pero este hecho no quiere decir que quienes parecieran no poseer este talento en las artes no puedan hacer ilustraciones sencillas, pero con un impacto visual evidente. Y una de estas áreas del arte en las que hoy en día es posible desempeñarse con ayuda de la tecnología es sin duda el dibujo, debido a la existencia de tecnología como: tabletas gráficas con guías en un ordenador, o moldes creados a partir de impresoras que crean modelos tridimensionales que pueden ayudar a crear moldes para dibujar un sin fin de figuras que se pueden plasmar en papel. (Sanz, 2019). Inclusive la manera tradicional de reproducir dibujos es aún posible a la hora de copiar un modelo existente, como dibujar sobre una ventana o un panel de luz.

Por tanto, la facilidad que nos ofrecen en la actualidad las herramientas informáticas, permiten realizar, con un ordenador o un móvil, diferentes tipos de tareas, incluyendo la posibilidad de emprender en una disciplina artística como es el dibujo, es posible. Dejando a un lado los tutoriales que pretenden guiar a los usuarios en su camino del dibujo existen varias posibilidades con las nuevas tecnologías emergentes y con las que ya existían hace ya tiempo dándoles diferentes usos, enfocándonos en este caso al dibujo.

Una de las formas más conocidas para emprender el camino hacia el dibujo, tomando en consideración la tecnología, son los programas para dibujar que están disponibles tanto para ordenadores como para móviles. Estos productos software están diseñados para lograr armar un dibujo a partir de figuras prediseñadas, o de retocar una imagen quitándole el fondo y/o juntar otras más. (Marker, 2020). De esta forma es posible reproducir o copiar dibujos pre diseñados, así como se hace tradicionalmente al copiar sobre papel o sobre luz para obtener una reproducción lo más precisa posible, del dibujo original.

Existen aplicaciones móviles que ofrecen apoyo para dibujar, mostrando o señalando pasos específicos a seguir para lograr un dibujo en concreto. Si se siguen los pasos indicados de

manera análoga a las figuras ofrecidas por alguna de las aplicaciones móviles, es posible lograr un resultado visual resaltante, pero aun así esta ayuda puede no ser suficiente para los usuarios, debido que aún estos trazos sencillos a seguir suelen necesitar de tiempo para capturar una forma similar a la mostrada por las aplicaciones razón por la cual, los usuarios dejan de dibujar o pierden el interés.

1.3.1. PROBLEMA CENTRAL

¿Cómo plasmar bocetos sencillos, utilizando solo líneas, teniendo muy poca o ninguna habilidad en el dibujo?

1.3.2. PROBLEMAS SECUNDARIOS

- Poca habilidad para realizar bocetos o dibujos de representación con trazos.
- Pocos están dispuestos a tomar clases de arte o dibujo y seguir con el curso hasta obtener habilidades prácticas.
- El pensamiento de que para dibujar se requiere de talento.
- Aplicaciones para dibujar con coste muy alto o coste mensual.
- No existe mucho uso de realidad aumentada para crear arte.

1.4. DEFINICIÓN DE OBJETIVOS

1.4.1. OBJETIVO GENERAL

Desarrollar una aplicación con realidad aumentada y APIs de inteligencia artificial, que permita dibujar bocetos sencillos con solo líneas, aunque se tenga poca o ninguna habilidad en el dibujo.

1.4.2. OBJETIVOS ESPECÍFICOS

- Establecer la visualización de bocetos existentes en una hoja de papel con realidad aumentada.
- Establecer el uso de una API de inteligencia artificial para encontrar la semejanza entre imágenes, en este caso bocetos y compararlos.
- Designar el almacenamiento en la nube para la integración de la API con la aplicación web.
- Crear la interfaz necesaria para visualizar a través de la realidad aumentada un boceto para reproducir el mismo en papel.
- Integrar realidad aumentada y almacenamiento en la nube para mostrar diferentes bocetos.

1.5. HIPÓTESIS

El uso de la realidad aumentada en conjunto con una API de inteligencia artificial, hace posible dibujar bocetos sencillos con líneas, semejantes al modelo original con un error del 10%.

1.5.1. OPERACIONALIZACIÓN DE VARIABLES

- Variable independiente:

Realidad aumentada en conjunto con la API de inteligencia artificial.

- Variable dependiente:

Dibujo de bocetos sencillos con líneas.

1.6. JUSTIFICACIÓN

1.6.1. JUSTIFICACIÓN ECONÓMICA

La investigación está disponible para todo el público al igual que la aplicación y su código fuente para su constante mejora o aporte. Los bocetos son de carácter abierto para todo el público y pueden ser compartidos. Por lo que todo el contenido será de acceso libre.

1.6.2. JUSTIFICACIÓN SOCIAL

La presente investigación tiene por característica proveer una ayuda para dibujar bocetos sencillos con líneas de forma fácil y sencilla. Estando el mismo disponible para su constante ampliación en cantidad de dibujos.

1.6.3. JUSTIFICACIÓN CIENTÍFICA

La finalidad de esta investigación es explorar la realidad aumentada y su aplicabilidad en uno de los campos del arte, el cual es el dibujo, para permitir a un usuario reproducir un dibujo pre existente. Así como la utilización de herramientas como A-frame, almacenamiento en la nube, Django y procesamiento de imágenes.

1.7. ALCANCES Y LÍMITES

1.7.1. ALCANCES

- Una aplicación que utiliza realidad aumentada para reconocer un solo objetivo y superponer un dibujo sencillo con líneas.
- Dibujos almacenados en la nube como imágenes.
- Se compara la semejanza del dibujo con el original para obtener un porcentaje de semejanza.
- Utilización del framework Django juntamente con Ar.js, A-frame y almacenamiento en la nube para la utilización de realidad aumentada.

- Publicación de dibujos nuevos para ser renderizados en la vista de realidad aumentada.

1.7.2. LÍMITES

- La precisión de la superposición de los bocetos sobre la hoja de papel está limitada a la calidad de imagen percibida por la cámara.
- Los dibujos sólo son imágenes y no se puede guardar ningún progreso dentro de la aplicación.
- Los dibujos pueden ser comparados de acuerdo a un solo boceto prediseñado.
- La aplicación no garantiza el aprendizaje de dibujo se limita a la práctica del mismo dependiendo del usuario al seguir las instrucciones provistas.
- La aplicación está limitada a los requerimientos de las tecnologías empleadas.

1.8. METODOLOGÍA

La investigación presente estudia las características de la realidad aumentada, aplicada al arte, tomando parte en la realización de un boceto sencillo con líneas para comprender si es posible realizar esta tarea, con dichas tecnologías.

Como metodología de investigación se utilizó el método científico haciendo énfasis en la etapa de experimentación donde se lleva a cabo, en primera instancia, la metodología de desarrollo para integrar las tecnologías necesarias y realizar así pruebas para la demostración de la hipótesis, siguiendo con el análisis para llegando así a las conclusiones o resultados.

Como metodología de desarrollo se empleó scrum en la etapa experimentación de la metodología de investigación donde se realizó la aplicación web que permite dibujar bocetos sencillos con líneas y comparar estos bocetos con un boceto prediseñado, proveniente de un manual de dibujo. En el desarrollo de esta aplicación web se buscó

procesos y entregas adaptables con una mejor y retroalimentación continua, razón por la cual se hace uso de scrum. Así también se adicionará a scrum una metodología de desarrollo para realidad aumentada, para el propósito del presente trabajo, con la finalidad de analizar los procesos y problemas que se presentan en el desarrollo de una aplicación que utiliza como herramienta principal a la realidad aumentada.



2.1. INTRODUCCIÓN

En los siguientes puntos se describirán conceptos necesarios para el desarrollo del presente trabajo como ser conceptos básicos sobre lo que significa realizar un dibujo hasta el trazado de bocetos sencillos usando exclusivamente líneas y cómo esta tarea puede ser realizada con realidad aumentada. Además, también se tratarán conceptos sobre los cuales se realizará el proyecto como tecnologías de realidad aumentada, tecnologías de realidad aumentada en la web, desarrollo de una aplicación web frente a una página web y desarrollo web en Django. Así como también la metodología a implementar. Se utilizará el IDE de Visual Studio Code con extensiones para Django, Python, JavaScript y HTML.

2.2. EL DIBUJO

Como primer concepto, haciendo referencia al término de dibujo se tiene que este es una representación en proporción que debe tener en sus partes y medidas la figura del objeto que se intenta representar. (Diccionarios Everest, 2015). Por tanto, el dibujo es proyectar un objeto o una idea plasmando este trabajo en algún tipo de superficie que no necesariamente tiene que ser un lienzo de papel. Y este puede comenzar desde un boceto.

El dibujo tomado como boceto es fundamental y funciona como base para cualquier dibujo al estar conformado por líneas y esbozos sobre un lienzo. Es primordial sobre todo por su rápida y económica ejecución. Siendo este concepto de boceto útil para adquirir destreza visual y ordenar información. (Labiano, 2019)

Sin embargo, el dibujo es un término que define el lenguaje de numerosas disciplinas, más allá de su aspecto subjetivo, este es accesible a toda persona sin limitaciones de idioma, ni edad, o tiempo. Es por esta razón que se tomará al dibujo con las definiciones anteriores, tomadas desde el trazado de un boceto. (Labiano, 2019)

2.2.1. EL BOCETO

Un boceto, también denominado bosquejo, sin importar el medio (papel o soporte digital). El boceto se compone de los trazos previos o provisionales que se realizan antes de abarcar la obra artística, también se usa el boceto para presentar ideas de un proyecto a modo de esquemas o diagramas. (López, 2021)

2.2.2. UTILIDAD DEL BOCETO

El boceto es fundamental debido a su sencillez de llevar a cabo y resulta una técnica muy rápida para plasmar visualmente lo que tenemos en nuestra imaginación o mente. En general nos sirve para probar de la forma más rápida posible todas las ideas que se nos vayan ocurriendo. El uso del boceto nos puede servir para: tener una idea general, tener un modelo previo muy detallado. (López, 2021)

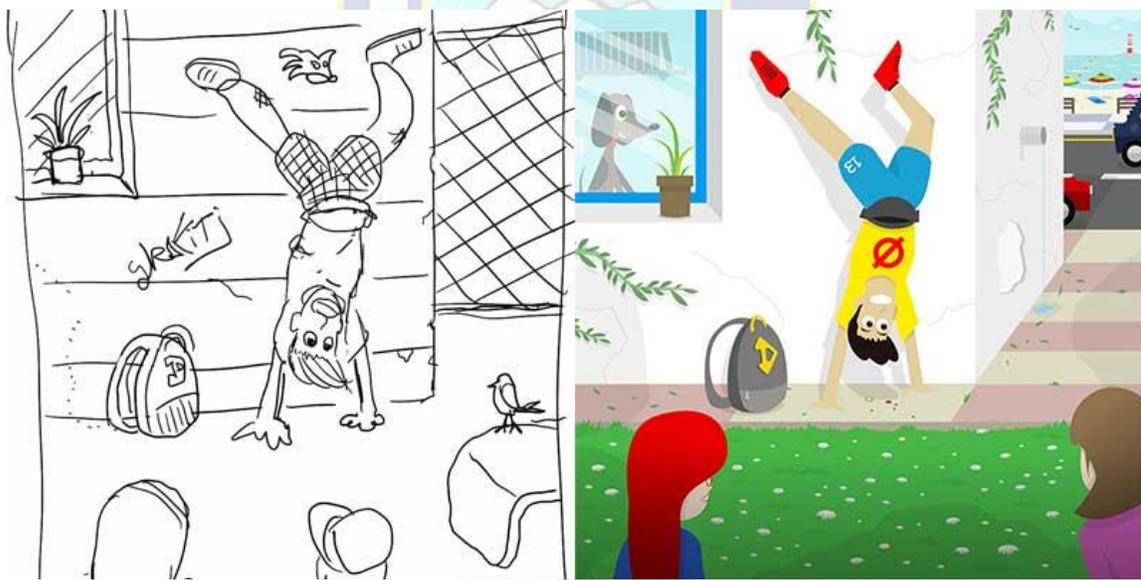


Figura 2.1. El boceto frente a un dibujo terminado

Fuente: (López, 2021)

2.2.3. VENTAJAS DEL BOCETO

Las ventajas realizar un boceto antes de ilustrar de forma más detallada al representar una idea, concepto u objeto, son: (López, 2021)

- Se requiere muy pocos recursos para realizar un boceto.
- El proceso de un boceto es muy rápido.
- Ayuda a mejorar la técnica de dibujo.
- Se cuenta con un modelo sólido para comenzar a diseñar o ilustrar.

2.3. REALIDAD AUMENTADA

Desde hace unos años las tecnologías de la información han rebasado las expectativas que se tenían y han cambiado la manera en la que vivimos y percibimos el mundo, como: la forma en la que nos comunicamos y relacionamos con otras personas, la forma en la que disfrutamos de los contenidos disponibles como los libros y películas. Entonces por lo tanto ha cambiado en gran manera la forma en la que percibimos el mundo haciendo que este cambie al observar a través de las tecnologías de información. Por lo tanto, el concepto de realidad aumentada puede partir de estos conceptos sencillos que está relacionado precisamente con esa última característica, que fue descrita sobre las tecnologías de la información, y es en cómo percibimos el mundo y la realidad aumentada nos ayuda a enriquecer nuestra percepción de la realidad. Por lo tanto, la realidad aumentada es una nueva ventana a través de la cual se puede ver el mundo, un mundo enriquecido. (Ariel, 2015).

2.3.1. FUNCIONAMIENTO BÁSICO

Para explicar de manera sencilla en qué consiste la realidad aumentada hay que hacer referencia a los sentidos humanos, haciendo hincapié en la percepción de la vista, a través de los cuales percibimos el mundo que nos rodea. Nuestra realidad física es entendida a través de la vista, el oído, el tacto y el gusto. La realidad aumentada viene a potenciar esos cinco sentidos, en especial la de la vista, superponiendo de distintas formas, información que se pueda percibir y elevar la realidad física con ciertos elementos que no están ahí. Utilizando superposición en tiempo real de: imágenes, marcadores, videos o cualquier tipo de información perceptible. Creando así información que se fusiona con la realidad, de tal

forma que el usuario llegará a pensar que la información superpuesta es parte del mundo físico.

Una descripción de alto nivel del funcionamiento de la realidad aumentada es mostrada en la figura adjunta. (Figura 2.1.) La tecnología funciona como un lente a través del cual vemos el mundo físico. La gran capacidad de esta lente, que el sistema de realidad aumentada, es la de superponer sobre el entorno físico información digital relevante con el contexto en el que se encuentra la persona observando, esta información es superpuesta a través de un marcador, para posicionarlo en la realidad. (Ariel, 2015).

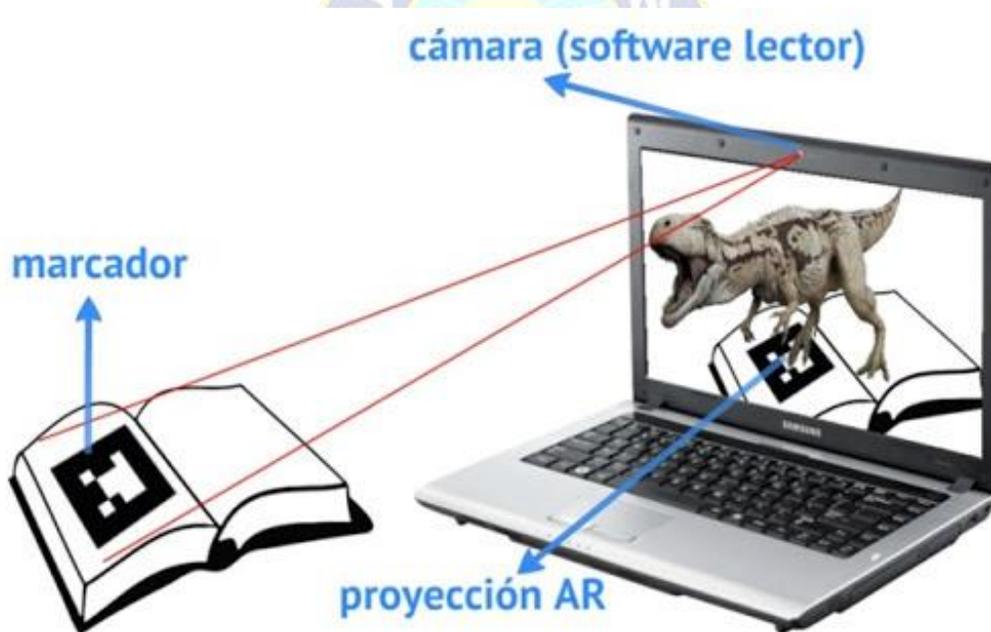


Figura 2.2. Esquema general del funcionamiento de la realidad aumentada

Fuente: (Fernández, 2020)

2.3.2. ELEMENTOS

- Primero se requiere de un elemento que capture las imágenes de la realidad, es decir que muestre lo que está viendo el usuario. Para esto se puede mencionar a la cámara como elemento para capturar las imágenes de la realidad.

- Por otro lado, se requiere el elemento sobre el cual se proyecta la información para sintetizar las imágenes reales con las superpuestas. Para esto se puede utilizar la pantalla de un ordenador, teléfono o consola de videojuegos.
- En tercer lugar, el elemento de procesamiento es necesario para interpretar la información del mundo real que recibe el usuario, generar la información virtual que cada servicio concreto necesite y mezclar la información adecuada para así superponer la información agregada. Estos elementos se pueden encontrar en servicios o paquetes de información.
- Finalmente se requiere el elemento que se usará como marcador para que resulte sencillo al interpretador del mundo real captar los elementos y superponerlos de manera que se pueda posicionar y fijar la información de manera efectiva dando la impresión de que dicha información pertenece a la realidad.

Gracias a los elementos mencionados se puede construir el concepto de realidad aumentada dando lugar así a la mencionada superposición de información para que el usuario pueda percibir la información agregada y la información real física como una sola.

2.3.3. CARACTERÍSTICAS DE LA REALIDAD AUMENTADA

Se identifican tres características fundamentales para la realidad aumentada y su propia definición como un sistema de realidad aumentada, sin estar determinada a un hardware específico:

- Un sistema de realidad aumentada debe combinar realidad y virtualidad.
- Un sistema de realidad aumentada debe ser interactivo en tiempo real.
- La registración de un sistema de realidad aumentada debe ser de tipo tridimensional.

En la última definición, se presenta uno de los problemas básicos en realidad aumentada: los objetos virtuales tridimensionales y objetos reales deben mostrarse siempre alineados

entre sí. Se pueden utilizar varias técnicas para seguir la posición de los objetos virtuales y realizar una correcta colocación de ellos (seguimiento basado en reconocimiento de patrones a través de marcadores) para proporcionar diferentes niveles de precisión. El hardware que requiere esta técnica es por lo general más caro, pero la solución adoptada es más profesional. Y es que la precisión está dada por el tipo de tarea que se requiere realizar como la diferencia entre un sistema que deberá responder a una cirugía con una frecuencia de actualización mayor y, la simple colocación de un objeto sobre un marcador en una mesa. (Alcarria, 2010).

2.4. REALIDAD AUMENTADA EN LA WEB

Al ser la realidad aumentada capaz de superponer contenido virtual en el mundo real. Este puede ser provisto por muchos tipos de dispositivos como: dispositivos móviles, cascos, ordenadores, y muchos otros más. Para aquellos dispositivos que soportan la realidad aumentada son capaces de capturar la realidad por una o más cámaras, luego mostrar esto en alguna pantalla y añadir a este algún tipo de contenido que no pertenece a la realidad. (Carpignoli, 2020)

Para los desarrolladores la realidad aumentada en la web significa evitar todos los esfuerzos y costos que significan el desarrollo móvil y la publicación de esta. Esto también significa reusar tecnologías como: JavaScript, HTML, y CSS, que muchos desarrolladores conocen. Haciendo posible que la realidad aumentada esté disponible para una amplia gama de dispositivos no solo móviles sino también ordenadores que cuenten con los requisitos necesarios para trabajar con esta tecnología. (Carpignoli, 2020)

Esto también significa que es posible arreglar problemas instantáneamente y desplegar cambios instantáneamente en la página web, abriendo así muchas posibilidades prácticas sobre esta tecnología. (Carpignoli, 2020)

Para los usuarios significa alcanzar una experiencia de realidad aumentada por tan solo visitar una página web. Así como los códigos QR, esta tecnología es alcanzable tan solo con acceso a una dirección web. Adicionalmente los usuarios no tienen que tener espacio

adicional para apreciar la realidad aumentada, ni tampoco tienen que mantener ninguna aplicación actualizada. (Carpignoli, 2020)

2.4.1. A-FRAME

A-Frame es una librería construida para experiencias de realidad aumentada. Esta librería está pensada para ser usada sobre HTML, haciendo su uso simple. Sin embargo, A-Frame no es solo una herramienta que provee y es capaz de renderizar objetos en tres dimensiones. Su núcleo bajo un marco de entidad-componente que proporciona una estructura declarativa y extensible compatible con muchas otras librerías escritas en JavaScript. (Carpenter, 2020)

Originalmente A-Frame fue mantenida dentro de Mozilla y ahora es mantenida dentro de Supermedium, A-Frame fue desarrollada con el objetivo de proveer una manera sencilla de proveer contenido sobre realidad virtual, como un proyecto de código abierto. (Carpenter, 2020)

A-Frame fue pensado para la creación de contenido tridimensional y de realidad virtual con HTML sin pasos de construcción, todo en la web. Permite una configuración de escena con HTML para manejar la renderización del lienzo, como luces, posición y rotación de escenario. Tiene compatibilidad con la mayoría de bibliotecas y marcos Web existentes. Su arquitectura de componente - entidad favorece la composición sobre la herencia para definir objetos tridimensionales utilizando componentes reutilizables. Está totalmente construida sobre JavaScript por lo que es compatible con casi cualquier librería de desarrollo web. Se ejecuta directamente en el navegador. (Carpenter, 2020)

2.4.2. AR.js

AR.js es una librería que dispone al programador con todas las herramientas para desarrollar un producto en la web con realidad aumentada. Al ser la web considerada un ambiente colaborativo y accesible. También es posible tomar a la realidad aumentada como

un medio de comunicación disponible en la web para su muestra y desarrollo abierto. (Carpignoli, 2020)

Esta librería está basada en A-Frame para renderizar y superponer contenido a través de la cámara de algún dispositivo. Además, trabaja juntamente con otra librería de código abierto basadas en JavaScript para proveer diferentes funciones a la librería. (Carpignoli, 2020)

2.4.2.1. TIPOS DE REALIDAD AUMENTADA

AR.js presenta los siguientes tipos de realidad aumentada en la web, estos tipos de realidad aumentada a pesar de tener un propósito estándar no están sujetos a un único uso, estos son:

- Seguimiento de imágenes. Cuando una imagen en dos dimensiones es encontrada por la cámara, es posible mostrar algún tipo de información sobre esta, como también cerca de esta. El contenido mostrado puede ser variado como, por ejemplo, es posible mostrar: otra imagen en dos dimensiones, una imagen animada en formato .gif, un modelo en tres dimensiones e inclusive un video plano dependiendo del soporte que tiene el navegador sobre el formato del video. Algunos casos de uso idóneos para este tipo de realidad aumentada con AR.js son: realidad aumentada en los cuadros artísticos, realidad aumentada en libros ilustrados, realidad aumentada en volantes, y publicidad. (Carpignoli, 2020)
- Realidad aumentada basada en la ubicación. Esta clase de realidad aumentada utiliza ubicaciones reales en el mundo para mostrar contenido y aumentar la realidad en el dispositivo del usuario. Las experiencias que pueden ser construidas sobre esta librería son aquellas que necesitan la posición real del usuario en el mundo. El usuario puede moverse y a través de algún dispositivo este podrá apreciar contenido con realidad aumentada en lugares reales sobre el mundo. Los objetos aparecerán de diferentes tamaños dependiendo de donde se encuentre el usuario sobre el mapa utilizando también, si es el caso, otras tecnologías para conocer la posición y rotación del dispositivo empleado por el usuario. Con esta solución es posible construir experiencias que den interactividad a destinos turísticos, o al

explorar lugares en una ciudad. También puede ser útil para encontrar lugares como: edificios, museos, restaurantes y hoteles. También es posible construir experiencias de aprendizaje utilizando la posición del usuario sobre el planeta. (Carpignoli, 2020)

- Seguimiento de marcadores. Cuando un marcador específico es encontrado por la cámara es posible desplegar contenido de la misma forma cómo funciona el seguimiento de imágenes. La diferencia principal con el seguimiento de imágenes es la estabilidad que provee un marcador, sin embargo, estos son limitados en tamaño y color. Este tipo de realidad aumentada es recomendable para experiencias que requieran de mucho contenido y contenido variado. También es útil cuando se requiere que el contenido mantenga cierta estabilidad al capturar un marcador. (Carpignoli, 2020)

2.4.2.2. REQUERIMIENTOS

Algunos requerimientos y limitaciones sobre AR.js están listadas a continuación:

- AR.js trabaja en cualquier móvil con soporte para WebGL y WebRTC.
- El seguimiento de marcadores no requiere muchos recursos físicos, sin embargo, el seguimiento de imágenes si requiere de más recursos.
- No es posible trabajar con Chrome en iOS. Por el momento Chrome en iOS no tiene acceso a la cámara del dispositivo.
- En dispositivos con más de una cámara posterior pueden existir errores al momento de detectar la cámara a ser usada.
- Para trabajar con realidad aumentada basada en la ubicación el dispositivo requiere un sensor de GPS.

- Los dispositivos iOS requieren activación manual del GPS para trabajar con realidad aumentada basada en la ubicación.

Los requerimientos descritos anteriormente pueden cambiar dependiendo de las actualizaciones existentes sobre los dispositivos en el mercado. (Carpignoli, 2020)

2.4.2.3. DESPLIEGUE

Para acceder a la cámara o a los sensores de GPS de los diferentes dispositivos, y debido a restricciones de la mayoría de los navegadores de internet, se recomienda trabajar bajo un protocolo seguro de transferencia de hipertexto o https. (Carpignoli, 2020)

Todos los despliegues que vayan a ser efectuados usando la librería AR.js necesitan ser desplegados en un servidor. Es posible usar servidores locales estáticos para desplegar la aplicación web con AR.js. (Carpignoli, 2020)

2.5. APLICACIONES WEB Y SITIOS WEB

Los términos aplicación web y sitio web con frecuencia son utilizados sin distinción. Ambos términos suelen ser considerados iguales o con la misma función. Sin embargo, se trata de plataformas digitales con propósitos específicos que responden a necesidades muy diferentes. Cada uno de estos términos hace referencia a un conjunto distinto de funcionalidades que pueden ser ofrecidas, y dependen ampliamente de los objetivos del proyecto. (Pizarro, 2016)

Es así que la decisión de desarrollar alguna de estas plataformas depende de los productos y servicios que se quieran poner a disposición de los usuarios y los objetivos que se quieran alcanzar en los distintos canales digitales. Otro punto a tomar en cuenta al momento de tomar la decisión de optar por un sitio web o una aplicación web es el costo, así como el tiempo de desarrollo de un proyecto digital. Las cargas de trabajo asociadas a las distintas áreas varían en el desarrollo de una plataforma u otra, afectando la planificación de las tareas. Conocer las diferencias entre un sitio web y una aplicación web permitirá tomar las decisiones adecuadas para responder a un problema de manera adecuada. (Pizarro, 2016)

En pocas palabras, un sitio web es un conjunto de páginas estáticas que entregan información. Por su parte, las aplicaciones web son plataformas principalmente interactivas que se centran en que los usuarios realicen acciones y que la aplicación web responda con alguna otra acción. Esta diferencia es fundamental determina sus principales objetivos, características y funciones. (Pizarro, 2016)

2.5.1. APLICACIÓN WEB

El objetivo principal de una aplicación web es que el usuario realice una tarea o acción. También pueden entenderse como un programa que se utiliza desde el navegador que cumple o provee servicios específicos. Siendo una aplicación web el negocio o parte del negocio. Para su creación se usa CSS, HTML, y JavaScript complementariamente se pueden utilizar Frameworks de trabajo o software gratuitos de fuente abierta. Un ejemplo claro de aplicación web es google Docs que nos permite no solo editar documentos con distintos estilos sino también nos permite compartirlos a través de la misma web. (Pizarro, 2016)

2.5.2. SITIO WEB

El objetivo principal de un sitio web es compartir información y entregarla. Por lo tanto, consumir información es la tarea principal que realizan los usuarios al ingresar a un sitio de este tipo. Esta idea puede ser confundida, debido a que los sitios web por lo general contienen algún llamado a la acción adicional, como realizar un comentario o suscribirse a contenido adicional. La diferencia está en que estas funciones representan una parte pequeña del objetivo del sitio y no el objetivo principal del negocio. Es así que un sitio web no siempre es el negocio y es más comúnmente solo parte del negocio con el objetivo de proveer información del tipo necesario. Los sitios web además suelen comprender varias páginas interconectadas y requieren un gestor de contenido. (Pizarro, 2016)

2.5.3. PRINCIPALES DIFERENCIAS

Las principales diferencias que deben ser tomadas en cuenta al momento de optar por una de estos tipos de sitios son:

Tabla 2.1. Diferencias entre aplicación y sitio web

SITIO WEB	APLICACIÓN WEB
Conjunto de páginas web estructuradas en un dominio.	Software desarrollado con tecnologías web.
Informa lo objetivos del proyecto, empresa o negocio.	Es el núcleo del proyecto empresa o negocio.
Se puede crear con o sin código (constructores en línea)	Se crea con código (lenguajes, librerías, Frameworks)
Pueden ser creadas tanto por programadores como también por personas que no saben programar	Son creadas por programadores web profesionales o con experiencia.

Fuente: (EDteam, 2020)

2.6. FRAMEWORK

El concepto de framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de videojuegos, y para distintos otros ámbitos que requieran un desarrollo completo y rápido, debido a que un framework como cualquier otra tecnología también uno de sus propósitos es ayudar en alguna forma al desarrollo. (Gutiérrez, 2018)

En general, con el término framework, no estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Los objetivos principales en el empleo de un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones de diseño. (Gutiérrez, 2018)

Por tanto, un framework podría ser definido como el espacio de trabajo del que se dispone donde este espacio contiene un conjunto de componentes para un diseño reutilizable que facilita y agiliza el desarrollo de sistemas Web. Al final se trata de un conjunto de herramientas y módulos que se puede reutilizar para varios proyectos. Su uso se centra principalmente en el desarrollo de aplicaciones web. Gracias a esta estructura se puede: (Gutiérrez, 2018)

- Ahorrar tiempo creando un marco de trabajo prediseñado.
- Facilita la colaboración entre programadores, estandarizando los códigos y permitiendo trabajos colaborativos.
- Existen herramientas y otras utilidades que han sido diseñadas especialmente para ciertos frameworks.
- Evita la escritura de código repetitivo como, por ejemplo, el acceso a las bases de datos o la validación de formularios.

2.7. DJANGO

Django es un framework web de alto nivel que se centra en el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, y gracias a esto Django trata de integrar todo lo necesario para desarrollar una aplicación web. Es gratuito y de código abierto, además tiene soporte gratuito y de pago. (MDN, 2020)

2.7.1. CARACTERÍSTICAS

- Django sigue una filosofía comúnmente conocida como “Baterías incluidas” lo que significa que es capaz de proveer o integrar todo lo necesario para el desarrollo de una aplicación web. Logrando así que todos los componentes de una aplicación web se encuentren en un único producto. También es capaz de seguir principios de diseño consistentes. (MDN, 2020)
- Django puede ser y ha sido usado para construir casi cualquier tipo de sitio o aplicación web, desde sistemas que manejan contenidos e información específica, hasta redes sociales y sitios complejos que proveen algún tipo de servicio. Puede funcionar con cualquier framework o librería en el lado del cliente, y puede proveer contenido en casi cualquier formato, incluyendo: HTML, JSON, XML, y otros. (MDN, 2020)
- Django coadyuva la necesidad de proveer seguridad a un sitio o aplicación web automáticamente. Por ejemplo, Django, proporciona una manera segura de administrar cuentas de usuario y contraseñas, evitando así errores comunes como: colocar informaciones de sesión en cookies trabajando directamente con la base de datos, o se almacenan directamente las contraseñas en un hash de contraseñas. (MDN, 2020)
- Un hash de contraseña es un valor de longitud fija creado al enviar la contraseña a una función criptográfica. Django puede validar si la contraseña ingresada es correcta enviándola a través de una función hash y comparando la salida con el valor hash almacenado. Sin embargo, debido a la naturaleza unidireccional de la función, incluso si un valor hash almacenado se ve comprometido es difícil para un atacante resolver la contraseña original. Así Django permite la protección contra algunas vulnerabilidades de forma predeterminada incluida la falsificación de solicitudes entre sitios. (MDN, 2020)

- Django usa un componente basado en la arquitectura “sin compartir”, esto quiere decir que cada parte de la arquitectura es independiente de la otra, y por lo tanto puede ser reemplazada o cambiada si es necesario. Teniendo en cuenta la separación existente entre las partes significa que puede escalar para aumentar el tráfico al agregar hardware en cualquier nivel. (MDN, 2020)
- El código de Django está escrito usando principios y patrones de diseño para fomentar la creación de código mantenible y reutilizable. DRY significa que no repitas lo que hiciste, reduciendo así la cantidad de código. Django también promueve la agrupación de la funcionalidad relacionada en aplicaciones reutilizables y en un nivel más bajo. (MDN, 2020)
- Django está escrito en Python, el cual ejecuta en muchas plataformas. Lo que significa que no está sujeto a ninguna plataforma en particular, y puede ejecutar sus aplicaciones en muchas distribuciones de Linux, Windows y Mac OS X. Además, Django cuenta con el respaldo de muchos proveedores de alojamiento web, y que a menudo proporcionan una infraestructura específica y documentación para el alojamiento de sitios Django. (MDN, 2020)

2.7.2. FRAMEWORK DOGMÁTICO

Los frameworks dogmáticos son aquellos que opinan acerca de la “manera correcta” de gestionar cualquier tarea en particular. Ofrecen soporte para el desarrollo rápido en un dominio en particular, porque la manera correcta de hacer cualquier cosa está generalmente bien comprendida y bien documentada. (MDN, 2020)

Sin embargo, pueden ser menos flexibles para resolver problemas fuera de su dominio principal, y tienden a ofrecer menos opciones para elegir qué componentes y enfoques pueden usarse. Los frameworks no dogmáticos, al contrario, tienen muchas menos restricciones sobre el modo mejor de unir componentes para alcanzar un objetivo, o incluso qué componentes deberían usarse. Ofrecen más libertad a los desarrolladores al momento de usar las herramientas más adecuadas para completar una tarea en particular, sin

embargo, el desarrollador o desarrolladores tienen que encontrar estos componentes o soluciones. (MDN, 2020)

Django es por lo tanto dogmático, pero no del todo, porque ofrece cierta libertad al momento de construir un sitio o aplicación web. Proporciona un conjunto de componentes para gestionar la mayoría de las tareas de desarrollo web, sin embargo, la arquitectura desacoplada de Django implica que es posible elegir y seleccionar de entre numerosas opciones diferentes o añadir soporte para otras tareas completamente nuevas. (MDN, 2020)

2.7.3. ESTRUCTURA TRADICIONAL DE UN PROYECTO

En un sitio web tradicional basado en datos, una aplicación web espera peticiones HTTP del explorador web o de otro cliente. Cuando se recibe una petición la aplicación elabora lo que se necesita basándose en la URL y posiblemente en la información incluida en los datos POST o GET. Dependiendo de qué se necesita quizás pueda entonces leer o escribir información desde una base de datos o realizar otras tareas requeridas para satisfacer la petición. La aplicación devolverá a continuación una respuesta al explorador web, con frecuencia creando dinámicamente una página HTML para que el explorador la presente insertando los datos recuperados en marcadores de posición dentro de una plantilla HTML. Las aplicaciones web de Django normalmente agrupan el código que gestiona cada uno de estos pasos en ficheros separados. (MDN, 2020)

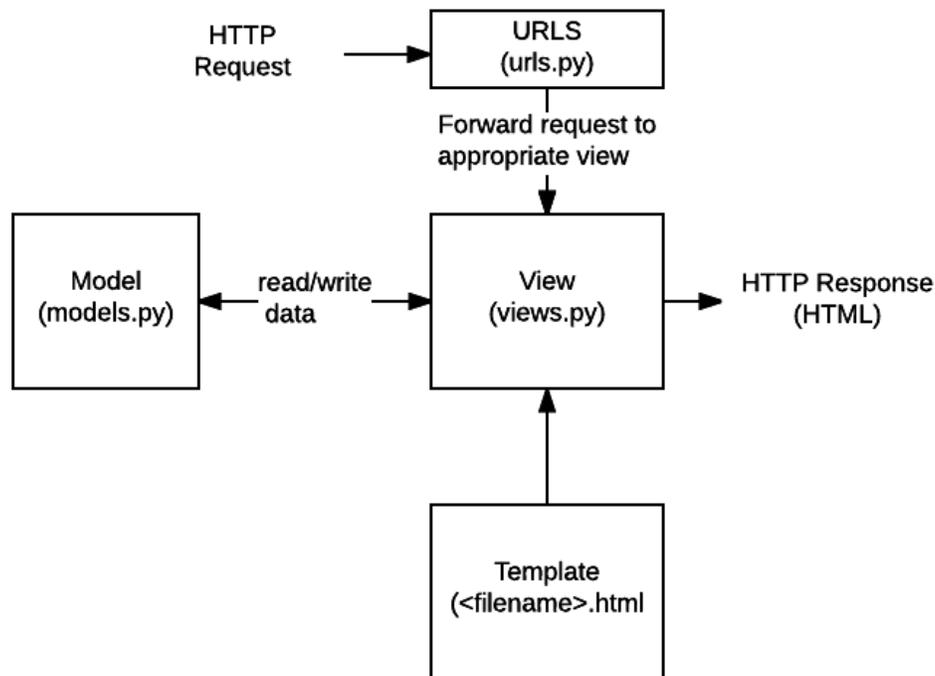


Figura 2.3. Pasos para mostrar una página

Fuente: (MDN, 2020)

- URLs: Aunque es posible procesar peticiones de cada URL individual vía una función individual, es mucho más sostenible escribir una función de visualización separada para cada recurso. Se usa un mapeador URL para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL se usa para redirigir las peticiones HTTP a la vista apropiada basándose en la URL de la petición. El mapeador URL puede también emparejar patrones de cadenas o dígitos específicos que aparecen en una URL y pasan a la función de visualización como datos. (MDN, 2020)
- Vista: Una vista es una función de gestión de peticiones que recibe peticiones HTTP y devuelve respuestas HTTP. Las vistas acceden a los datos que necesitan para satisfacer las peticiones por medio de modelos, y delegan el formateo de la respuesta a las plantillas. (MDN, 2020)

- Modelos: Los Modelos son objetos de Python que definen la estructura de los datos de una aplicación y proporcionan mecanismos para gestionar (añadir, modificar y borrar) y consultar registros en la base de datos. (MDN, 2020)
- Plantillas: una plantilla (template) es un fichero de texto que define la estructura o diagrama de otro fichero (tal como una página HTML), con marcadores de posición que se utilizan para representar el contenido real. Una vista puede crear dinámicamente una página usando una plantilla, llenándola con datos de un modelo. Una plantilla se puede usar para definir la estructura de cualquier tipo de fichero. (MDN, 2020)

2.7.4. PATRÓN DE DISEÑO

Django sigue el patrón MVC que significa, modelo, vista y controlador. Sin embargo, en Django MVC tiene un concepto con ciertas diferencias dentro del framework mismo: (Django Documentation, 2020)

- Modelo, la porción de acceso a la base de datos, es manejada por la capa de la base de datos de Django. (Django Documentation, 2020)
- Vista, la porción que selecciona qué datos mostrar cómo mostrarlos, es manejada por la vista y las plantillas. (Django Documentation, 2020)
- Controlador, la porción que delega a la vista dependiendo de la entrada del usuario, es manejada por el framework mismo siguiendo una configuración URL y llamando a la función apropiada de Python para la URL obtenida. (Django Documentation, 2020)

Debido a que el controlador es manejado por el mismo Django y la parte más importante se produce en los modelos, las plantillas y las vistas, Django es conocido como un Framework MTV. En el patrón de diseño MTV, modelo, template, vista. (Django Documentation, 2020)

- Modelo, la capa de acceso a la base de datos. Esta capa contiene toda la información sobre los datos: cómo acceder a estos, cómo validarlos, cuál es el comportamiento que tiene, y las relaciones entre los datos. (Django Documentation, 2020)
- Template, la capa de presentación. Esta capa contiene las decisiones relacionadas a la presentación: como algunas cosas son mostradas sobre una página web u otro tipo de documento. (Django Documentation, 2020)
- Vista, la capa de la lógica de negocios. Esta capa contiene la lógica que accede al modelo y la delega a la plantilla apropiada. Es posible pensar en esto como un puente entre la capa de modelo y la capa de template o plantilla. (Django Documentation, 2020)

Dicho de otra forma, se puede considerar que las vistas de Django pueden ser el controlador y las plantillas o templates de Django pueden ser la vista. En la interpretación de Django MVC, la vista describe los datos que son presentados al usuario, no necesariamente el cómo será mostrado, pero sí cuáles datos son presentados. En contraste con otros frameworks similares sugieren que el trabajo del controlador incluya la decisión de cuales datos son presentados al usuario, mientras que la vista se estrictamente el cómo serán presentados y no cuáles. (Django Documentation, 2020)

2.8. API

En términos más generales, una API es una interfaz de programación de aplicaciones, es decir, es un conjunto de funciones, procedimientos o clases que un sistema operativo, librería o servicio proporciona para soportar peticiones realizadas por un software. Permite que las funcionalidades de un servicio inicial puedan ser ampliadas a otras, pudiendo integrar otras APIs haciendo que el servicio proporcionado por el software sea más extensible.

En conclusión, las APIs facilitan considerablemente la construcción de aplicaciones más complejas; un aspecto que estimula a una nueva generación de soluciones de tipo, respuesta rápida, y aplicaciones cliente que operan en sistemas de mayor extensión. (Soto, 2018)

2.8.1. API REST

Una API REST permite la utilización de servicios web disponibles a través de Internet utilizando el protocolo estándar HTTP; en otras palabras, una API REST es un servicio que al igual que el término inicial de una API provee de funciones que dan capacidad de usar un servicio web de terceros dentro de nuestro propio servicio o aplicación propia de un modo seguro. (Soto, 2018)



Figura 2.4. Servicio de un API REST

Fuente: (Soto, 2018)

Mediante un método GET se obtiene una respuesta a una consulta en una API REST. El contenido de las respuestas son datos en diferentes formatos como pueden ser: texto plano, ficheros XML, JSON CSV. (Soto, 2018)

Los servicios de una API REST son los intermediarios entre el contenedor de los datos con la parte, que sería la que interactúa con el usuario final. (Soto, 2018)

2.9. INTELIGENCIA ARTIFICIAL

El término inteligencia artificial mueve las emociones de quienes lo escuchan. Por una cosa, la fascinación con la inteligencia, lo que aparentemente imparte a los humanos un lugar especial entre las formas de vida. Surgen preguntas, tales como: ¿Qué es inteligencia?, ¿Cómo uno puede medir la inteligencia?, o ¿Cómo trabaja el cerebro? Todas estas preguntas son importantes al tratar de entender inteligencia artificial. Sin embargo, la pregunta más importante es cómo una máquina inteligente puede tener un comportamiento similar al de una persona mostrando, en ciertos aspectos, una conducta inteligente. (Wolfgang, 2017)

El atributo artificial puede despertar diferentes asociaciones. Nos trae a la mente novelas de ciencia ficción y la pregunta de si se puede imitar totalmente el comportamiento humano desde el lado racional hasta el lado irracional. (Wolfgang, 2017)

Con tantas interpretaciones es difícil definir el término de inteligencia artificial de manera robusta. Sin embargo, es posible recurrir a las caracterizaciones que tomó el campo de la inteligencia artificial. En 1995, Jhon McCarthy, uno de los pioneros de la inteligencia artificial, fue el primero en definir el término de inteligencia artificial, aproximadamente como sigue: El objetivo de la inteligencia artificial es desarrollar máquinas que se comporten como si fuesen inteligentes. Por otro lado, esta definición es insuficiente porque la inteligencia artificial tiene como meta resolver problemas prácticos difíciles que pueden ser demasiado demandantes inclusive para el cerebro humano. La enciclopedia británica define a la inteligencia artificial como, la habilidad de las computadoras digitales o computadoras controladas por robots para resolver problemas que están normalmente asociados con las capacidades intelectuales de los humanos para procesarlos. Pero esta definición también tiene debilidades. Puede admitir que por ejemplo una computadora puede tener una gran memoria con la capacidad de guardar textos largos y recuperar toda esa información de manera que se pueda demostrar capacidades inteligentes que pueden parecer realizadas por memorización. Este problema es resuelto elegantemente por Elaine Rich quien define a la inteligencia artificial como, el estudio de cómo hacer que las

computadoras realicen tareas en el momento en el que las personas son mejores. (Wolfgang, 2017)

Por lo tanto, la inteligencia artificial tiene por objetivo que los ordenadores hagan la misma clase de cosas que puede hacer el cerebro humano representando así la automatización de distintas tareas. Algunas como razonar suelen describirse como inteligentes. Otras como la visión, no. Pero todas entrañan competencias psicológicas que permiten a los seres humanos alcanzar sus objetivos. La inteligencia no es una dimensión única, sino un espacio profusamente estructurado de capacidades diversas para procesar la información. Del mismo modo, la inteligencia artificial utiliza muchas técnicas diferentes para resolver una gran variedad de tareas. (Boden, 2017)

2.9.1. TIPOS DE INTELIGENCIA ARTIFICIAL

Tomando en cuenta que la inteligencia artificial es la capacidad de una máquina para imitar el comportamiento humano a través de algoritmos sacando provecho de los datos y el poder computacional, se puede clasificar en los siguientes tipos:

2.9.1.1. INTELIGENCIA ARTIFICIAL ESTRECHA

Realiza tareas específicas como reconocer un rostro o una voz. También es común en problemas de clasificación. (Zerna, 2020)

2.9.1.2. INTELIGENCIA ARTIFICIAL GENERAL

Es similar a la conducta humana, toma el conocimiento en lo que ha sido entrenado y aprende algo nuevo. Un ejemplo claro de esta aplicación son los asistentes de voz que aparecen en el mercado de los móviles. (Zerna, 2020)

2.9.1.3. SUPER INTELIGENCIA ARTIFICIAL

Es un proyecto que permanece en la fantasía, pero se cree que superará la inteligencia de los humanos. (Zerna, 2020)

Cabe resaltar que la inteligencia artificial no es como el aprendizaje automático. La inteligencia artificial es la capacidad de las computadoras de mostrar un comportamiento inteligente. El aprendizaje automático es la técnica para mejorar dicho comportamiento. (Zerna, 2020)

2.9.2. MEDIDA DEL ÍNDICE DE SIMILITUD ESTRUCTURAL

La medida del índice de similitud estructural (SSIM), iniciada primero por Alan Bovik en el 2001 y después estudiada y publicada por la Universidad de Waterloo, es un método para predecir la calidad percibida de las imágenes o videos digitales. SSIM se utiliza, por tanto, para medir la similitud entre dos imágenes. El índice SSIM es una métrica de referencia completa, en otras palabras, la medición o predicción de la calidad de la imagen se basa en una imagen inicial sin comprimir o sin distorsiones como referencia. (Rosebrock, 2021)

La medida del índice de similitud estructural es un modelo basado en la percepción que considera la degradación de la imagen como un cambio percibido en la información estructural, al tiempo que incorpora importantes fenómenos de percepción, incluidos los términos de enmascaramiento de luminancia y enmascaramiento de contraste. La diferencia con otras técnicas como MSE o error medio cuadrado es que estos enfoques estiman errores absolutos. La información estructural es la idea de que los píxeles tienen fuertes interdependencias, especialmente cuando están espacialmente cerca. Estas dependencias llevan información importante sobre la estructura de los objetos en la escena visual. (Rosebrock, 2021)

Este índice tiene como propósito medir la similitud entre dos imágenes, prediciendo mediante la medición o predicción la calidad de la imagen. Es así que se ha usado para predecir la depreciación de la calidad de una imagen, cambios visuales en fotografías sucesivas, o la similitud entre dos imágenes según su contexto. (Rosebrock, 2021)

2.9.2.1. ENTRADAS Y SALIDAS

Cuando un ordenador observa una imagen, verá una matriz de valores que representan píxeles. Dependiendo de la resolución o tamaño de la imagen, el ordenador verá una matriz de números $32 \times 32 \times 3$, donde el tres significan los valores en formato RGB. Por ejemplo, si una imagen en formato JPG de tamaño 480×480 . La matriz representativa sería una de $480 \times 480 \times 3$. Cada uno de estos campos en la matriz tiene un valor entre 0 y 255 que describe la intensidad de los píxeles en ese punto. Estos números, que no tienen significado cuando un individuo clasifica imágenes con su vista, son la única entrada disponible para un ordenador. La idea es que la computadora reciba esta matriz de números y la salida son números que representan la probabilidad de que una imagen sea de cierta clase. (Deshpande, 2016)



Figura 2.5. Representación de una imagen en un ordenador

Fuente: (Deshpande, 2016)

Una vez que ya se definieron las entradas y las salidas, lo que se quiere es que la computadora sea capaz de diferenciar entre todas las imágenes las características que hacen única a una imagen para clasificarla en una clase. (Deshpande, 2016)

2.9.2.2. ÍNDICE DE SIMILITUD ESTRUCTURAL

El algoritmo del índice de similitud estructural es calculado en varias ventanas de una imagen. La medida entre dos ventanas x y y de un mismo $N \times N$. (Rosebrock, 2021)

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Donde:

μ_x : el promedio de x

μ_y : el promedio de y

σ_x^2 : la varianza de x

σ_y^2 : la varianza de y

σ_{xy} : la covarianza de x y y

$c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ constantes

L : está en un rango dinámico que depende de los bits que ocupa un pixel ($2^{\#bits \text{ por pixel}} - 1$)

$k_1 = 0.01$ y $k_2 = 0.03$ por defecto (imagen referencia e imagen a comparar)

Ecuación 2.1. Índice de similitud estructural

Con este índice aplicado a dos imágenes en varias, pero iguales secciones para las dos imágenes comparadas, se obtiene un índice de similitud estructural el cual es promediado dando lugar a un único índice que representa la similitud entre esas dos imágenes completas. Esto significa que mientras mayor sea la resolución de las imágenes comparadas se obtendrán resultados más precisos. (Rosebrock, 2021)

2.10. MÉTODO CIENTÍFICO

El método científico es una técnica que permite llegar a un conocimiento que puede ser de consideración válida. El método científico consiste, por lo tanto, en una forma para aproximarse a una realidad. Lo anterior quiere decir que el método científico cumple con dos características fundamentales: primero es la falsabilidad, es la posibilidad de que lo que

se quiere demostrar o refutar pueda ser con el tiempo revaluado con más evidencia. Segundo es la reproductividad, esto quiere decir que lo que se está demostrando o refutando puede ser reproducido por otra persona obteniendo el mismo resultado, derivando así a la misma conclusión. (Westreicher, 2021). Razón por la cual se hace uso del método científico en el presente trabajo. Gracias a la falsabilidad existe la posibilidad de que los resultados presentados puedan ser revaluados o mejorados. La reproductividad permite que los resultados puedan ser reproducidos.

2.10.1. PASOS DEL MÉTODO CIENTÍFICO

El método científico contiene una serie de etapas, siendo estas las que se describen a continuación: (Westreicher, 2021)

- Observación: Es recoger información de la realidad, encontrar algún hecho de relevancia.
- Inducción: A partir de lo observado, se realizan preguntas para obtener una afirmación.
- Hipótesis: Se plantea una proposición que pueda explicar lo analizado en los pasos anteriores.
- Experimentación: Se intenta demostrar la hipótesis planteada con los elementos que el investigador tenga a su disposición. Es decir, que el estudio se hace bajo ciertas condiciones.
- Análisis: Se realizan cálculos para considerar la información, tratando de comprender los datos obtenidos en la fase de experimentación.
- Conclusión: A partir de los resultados de la experimentación se puede demostrar o refutar la hipótesis.



Figura 2.6. Pasos del método científico

(Westreicher, 2021)

2.11. INGENIERÍA DE SOFTWARE

La ingeniería de sistemas es una especialidad que consiste en sistemas, instrumentos y técnicas que se emplean en el desarrollo de los programas informáticos. La ingeniería de software, también hace posible la integración entre el análisis, los antecedentes de la situación, el bosquejo del proyecto, así como el desarrollo del mismo, las pruebas necesarias para comprobar su funcionamiento correcto y poner en funcionamiento el sistema desarrollado. Se debe señalar, que el desarrollo de sistemas informáticos va unido a lo que se conoce como el ciclo de vida de software que consiste en cuatro etapas que se conocen como: Concepción, elaboración, construcción y transición. La concepción determina la repercusión del proyecto y diseña el modelo de negocio. La elaboración hace referencia a la planificación del proyecto, especificando las características y apoya la

arquitectura. La construcción es la elaboración del producto. La transición es la entrega del producto final a los usuarios. (Fernandez, 2017)

2.11.1. SCRUM

Scrum es una metodología ágil para el desarrollo de software o la gestión de proyectos. Antes de la definición Scrum, se debe definir el concepto de ágil. El desarrollo de software ágil se define como: la manera apropiada de responder ante los cambios siendo este el motor principal del desarrollo ágil. Sin embargo, el desarrollo ágil es más que solo una respuesta efectiva ante el cambio, incluye además del desarrollo rápido, la filosofía que recomienda las estructuras de equipo y las actitudes que hacen más fácil la comunicación entre los miembros del equipo, gente de negocios, entre los ingenieros de software y sus gerentes entre otros. Pone énfasis en la entrega rápida de software funcional y resta importancia a los productos intermedios del trabajo, adoptando al cliente como parte del equipo de desarrollo. Para que el desarrollo ágil se haga efectivo es esencial que este se diseñe en forma que permita al equipo del proyecto adaptar las tareas y hacerlas directas, ejecutar la planeación de manera que se haga posible el entendimiento de la fluidez de un enfoque ágil del desarrollo. Poniendo énfasis en una estrategia de entrega incremental que haga trabajar al software tan rápido como sea posible para el cliente, según el tipo de producto y el ambiente de operación. (Fernandez, 2017)

2.11.1.1. CARACTERÍSTICAS

De todas las metodologías ágiles, Scrum es única porque introduce la idea del control empírico de los procesos. Esto significa que Scrum utiliza el progreso real de un proyecto para planificar y concertar los lanzamientos. En Scrum, los proyectos se dividen en ritmos de trabajo breves, conocidos como sprints. Normalmente, tienen una, dos o tres semanas de duración. Al final de cada sprint, el cliente y los miembros del equipo se reúnen para evaluar el progreso del proyecto y planear los siguientes pasos a seguir. Esto permite que la dirección del proyecto se ajuste o se reoriente una vez finalizado el trabajo, sin especulaciones ni predicciones. (Fernandez, 2017)

2.11.1.2. HITOS DE LA METODOLOGÍA DE TRABAJO SCRUM

En la Figura 2.5 se describen los hitos dentro del proceso Scrum. El desarrollo iterativo se realiza en un sprint, que contiene los siguientes eventos: sprint planning (planeación de sprint), daily meeting (reunión diaria), sprint review (sprint de revisión) y sprint retrospective (sprint en retrospectiva). (Abellán, 2020)



Figura 2.7. Hitos del proceso Scrum

Fuente: (Abellán, 2020)

El sprint es el corazón del Scrum, es el contenedor de los demás hitos del proceso Scrum. Todo lo que ocurre en una iteración para entregar valor está dentro de un sprint. La duración máxima es de un mes, el tiempo se determina en base al nivel de comunicación que el cliente quiere tener con el equipo. Los sprints largos pueden hacer que se pierda retroalimentación valiosa del cliente y poner en peligro el proyecto. (Abellán, 2020)

2.12. METODOLOGÍA PARA EL DESARROLLO REALIDAD AUMENTADA

Para el desarrollo de una aplicación con realidad aumentada se requiere tomar en cuenta ciertos aspectos que requieren atención especial y que no pueden ser manejado eficientemente por herramientas existentes o un desarrollo tradicional de un proyecto. Siendo así que se propone una metodología basada en cascada, que toma especial y única atención en la parte del proyecto que requiere realidad aumentada. (Kulas, 2018)

La razón por la que se requiere una metodología específica para el campo de la realidad aumentada es debido a que esta es inherentemente multimodal, lo que describe una experimentación con diferentes técnicas de interacción, por la misma naturaleza de la realidad aumentada como herramienta. Esto tiene que ser diseñado, implementado y evaluado.

El enfoque en el desarrollo de realidad aumentada requiere especial atención en múltiples problemas y aspectos, que son: (Kulas, 2018)

2.12.1. PROBLEMAS DE PROCESO

En un desarrollo tradicional, se puede seguir una metodología de tipo cascada, siguiendo la fase de diseño, la implementación y finalmente la fase de evaluación. Estas fases básicas e importantes, aunque son efectivas en la mayoría de los casos, la realidad aumentada requiere retroalimentación y dependencias sobre el ambiente no solo la interacción básica con el usuario. Sin embargo, este proceso de cascada puede ser efectivo si se tiene un espacio general predecible, pero lo que se requiere en herramientas como la realidad aumentada es mejora constante e inmediata sobre dependencias del ambiente y retroalimentación. (Kulas, 2018)

2.12.2. HERRAMIENTAS

Se propone tres fases importantes para el desarrollo con enfoque en la realidad aumentada:

Diseño, el diseño no solo debe ser mantenible y entendible, también debe ser acelerado, además de optimizado para el entorno en el que se mostrará. Su diseño debe ser acelerado por los cambios o adiciones que puedan existir, para esto se debe definir una herramienta para esta tarea. (Kulas, 2018)

Implementación, la implementación a la realidad aumentada se realiza antes de su aplicación en el proyecto principal generado pequeños prototipos a parte de las funciones básicas. (Kulas, 2018)

Evaluación, la evaluación es previa implementación a el proyecto principal, pero también en la misma. La evaluación toma en cuenta visualización, desempeño de figuras y usabilidad. (Kulas, 2018)

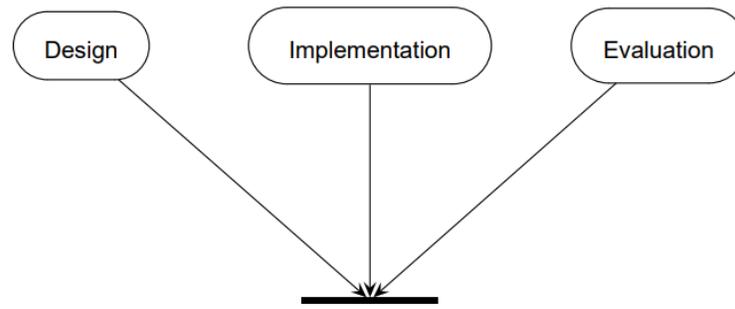


Figura 2.8. Fases para el desarrollo con realidad aumentada

Fuente: (Kulas, 2018)



3.1. INTRODUCCIÓN

El presente capítulo está dedicado al desarrollo de una aplicación web que manejará los datos necesarios para transformar imágenes en objetos que se puedan renderizar en una presentación con realidad aumentada basada en A-Frame con AR.js.

El desarrollo de la aplicación web estará sobre el Framework Django y será puesto a prueba en un servidor real, con el objetivo de lograr utilizar una API de comparación de imágenes con un puntaje. Para esto será necesario contar con servicio de computación en la nube, así como también de alojamiento en la nube para los archivos, ya que como se describió en el capítulo anterior es necesario desplegar los archivos en un servidor en la nube o local.

El uso de la metodología Scrum, contribuirá en gran medida en el desarrollo de la aplicación web, a través de cada iteración se podrá observar el avance, empezando desde la creación y configuración de servicios en la nube por Amazon hasta la implementación de la API y realidad aumentada sobre la aplicación web juntamente con el despliegue de toda la aplicación web.

Para el desarrollo de la interfaz de usuario se utilizará HTML, CSS, y JavaScript, así como algunas librerías externas como son: Google Fonts y Bootstrap. Siguiendo lineamientos simples de diseño.

Todo el desarrollo de la aplicación será puesto sobre el Framework Django, así como la implementación de la realidad aumentada, la API de comparación de imágenes y el manejo de archivos sobre servicios en la nube.

Para los requerimientos se tomaron en cuenta los de tipo usuario, siguiendo las limitaciones descritas por cada una de las herramientas a utilizar, así como de los objetivos de la aplicación web.

3.2. REALIDAD AUMENTADA CON AR.js

El tipo de realidad aumentada con AR.js que se usará en el presente proyecto será la realidad aumentada basada en marcadores, descrita en el capítulo anterior, este tipo de realidad aumentada requiere de marcadores específicos, con patrones específicos para ser reconocido por la cámara. También es necesario recordar que en este tipo de realidad aumentada es necesario contar un el marcador impreso y disponible para ser reconocido por la cámara, sin embargo, es recomendable tener impreso y también contar con iluminación suficiente para hacer visible a la cámara el marcador.



Figura 3.1. Un marcador por defecto conteniendo un código QR

Fuente: (Carpignoli, 2018)

En la Figura 3.1 se puede observar un marcador que AR.js tiene para usar instantáneamente para hacer pruebas por lo que es llamado un marcador por defecto llamado Hiro. Este marcador además está conteniendo un código QR. Sobre el marcador se puede apreciar a través de la cámara la realidad aumentada donde se está renderizando un objeto tridimensional.

3.3. METODOLOGÍA PROPUESTA

Como se ha mencionado la metodología a seguir son las etapas del método científico y la etapa correspondiente a este capítulo es la etapa de la experimentación, por lo que se llevará a cabo la metodología de experimentación bajo el marco de la metodología scrum para facilitar el trabajo de desarrollo de la aplicación web y así realizar las pruebas necesarias para la demostración de hipótesis. Por lo tanto, parte de la etapa de experimentación estará dedicada al desarrollo de la aplicación web paralelamente se realizarán pruebas gracias a la metodología scrum, por lo que se seguirá de cerca la etapa de experimentación en esta fase de desarrollo.

3.4. EXPERIMENTACIÓN

En esta etapa se pretende realizar el desarrollo de la aplicación web. Revisando y probando cada requerimiento gracias a la metodología scrum. En esta sección también está contemplada la parte de realidad aumentada siguiendo las fases propuestas por la metodología a usar propuesta anteriormente.

En primera instancia se continúa con el desarrollo de la metodología scrum:

3.4.1. PLANEACIÓN

Para la obtención de los requerimientos necesarios se recurrió a las siguientes tareas:

Información: Se realizó recopilación de información necesaria para la creación del contenido, analizando las herramientas a usar, así como también del comportamiento de las mismas y la distribución de su contenido.

Investigación: Se realizó investigación acerca de la calificación de una producción artística y dibujos en general, con enfoque a la precisión y su calificación sobre este parámetro. También se investigó sobre APIs disponibles en el análisis de contenido multimedia orientados a la comparación de imágenes.

Análisis comparativo: Se realizó comparación con otros usos de la realidad aumentada en el campo artístico, así como también de algunas aplicaciones similares en el área del dibujo con realidad aumentada y también la forma en la que los usuarios interactúan con estas otras aplicaciones diversas.

De acuerdo a la información obtenida, se diseña una aplicación web capaz de englobar la mayoría de sus usos en una sola página para evitar el movimiento en muchas páginas. También será necesario desarrollar el lado del administrador encargado de subir nuevos bocetos a la página. También será necesario realizar conexiones con Amazon Web Services para el alojamiento de los archivos estáticos de la aplicación. Considerando estos primeros aspectos se tiene:

- Preparar conexiones con servicios en la nube para alojar los archivos estáticos de la aplicación.
- Crear una estructura capaz de manejar administradores
- Crear formularios para administrar los dibujos disponibles para dibujar.
- Crear un formulario para la comparación de imágenes o fotografías.
- Llamar y emitir un resultado por parte de una API.
- Integrar la realidad aumentada con marcadores.

A continuación, siguiendo la tabla 3.1 se realizará la clasificación de los requerimientos, produciendo así el product backlog.

Tabla 3.1. Requerimientos

N	REQUERIMIENTOS
1	Diseño de una conexión a servicios en la nube
2	Administración de archivos dentro de la aplicación web
3	Manejo de formularios para administrar archivos
4	Acciones para el manejo de una API en la aplicación web
5	Integración de realidad aumentada

Fuente: Elaboración propia

A continuación, en la figura 3.2 se puede observar el proceso que la aplicación web realizará al final.



Figura 3.2. Estructura de la aplicación web

Fuente: Elaboración propia.

Es así que se define un tipo de actor que puede poseer uno o dos roles. A continuación, en la tabla 3.2 se detalla estos actores:

Tabla 3.2. Actores que intervienen en la aplicación web

ACTOR	ROL	TAREAS
Usuario	Administrador	<ul style="list-style-type: none">• Crear y publicar nuevos dibujos a través de Django.• Administrar los dibujos conjuntamente con los servicios de almacenamiento en la nube.
	Usuario simple	<ul style="list-style-type: none">• Acceder a los diferentes dibujos con realidad aumentada.• Comparar dibujos o fotografías a través del formulario de comparación.

Fuente: Elaboración propia

Ya establecidas todas las tareas que tendrá un usuario administrador, así como un usuario común, sigue el diagrama de caos de uso que contempla ambos tipos de usuarios. A continuación, en la figura 3.3 el caso de uso dentro de la aplicación web, mostrando cada una de las tareas que realizará la misma.

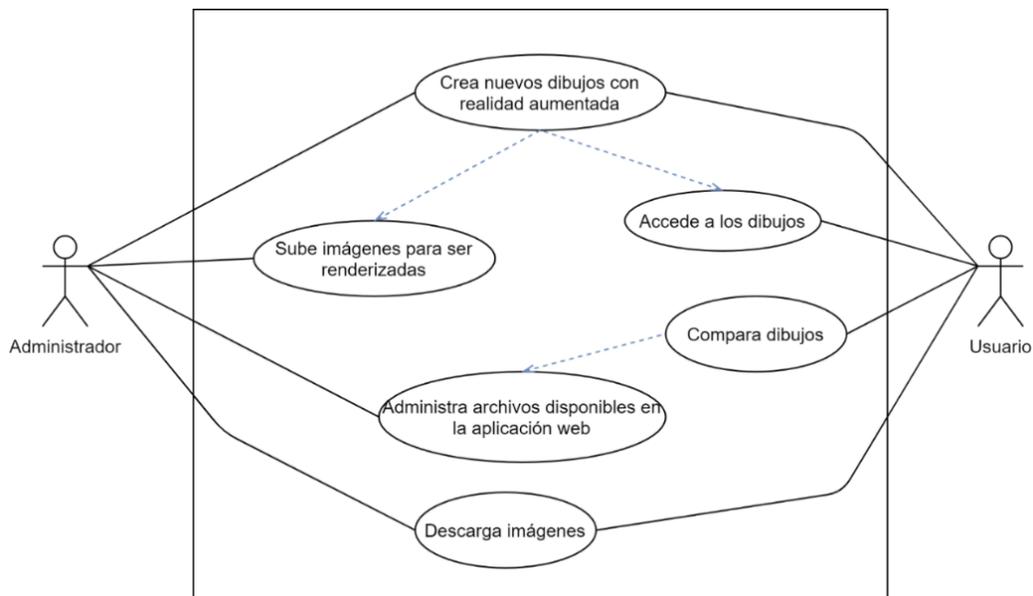


Figura 3.3. Diagrama de casos de uso de la aplicación web

Fuente: Elaboración propia

3.4.2. DESARROLLO

En esta etapa se desarrollará los Sprint correspondientes, de acuerdo a los requerimientos previamente establecidos.

3.4.2.1. SPRINT 1 CONEXIÓN A SERVICIOS EN LA NUBE

- Planeación: La conexión a servicios en la nube es importante para la utilización de APIs que requieren manejo de archivos en la nube, o como es el caso de la aplicación web, que requiere, además de archivos en la nube para la API, manejo de imágenes y archivos estáticos como Scripts de CSS y JavaScript.

También se realiza el análisis, sobre el diseño de algunas de las funciones del Administrador dentro de la aplicación web, que son acciones inmediatas al momento de requerir subir más imágenes para ser renderizadas. Es necesario también establecer una conexión para archivos estáticos debido a que los servicios de alojamiento de aplicaciones web, frecuentemente no soportan la carga y conexión de archivos estáticos. Figura 3.4.

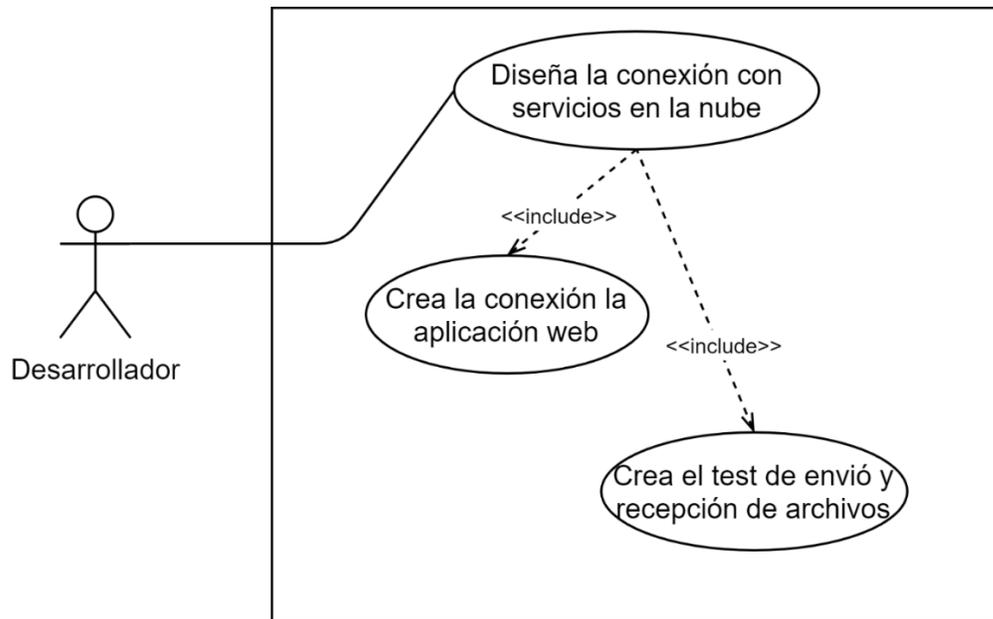


Figura 3.4. Diagrama de casos de uso – Desarrollador

Fuente: Elaboración propia

En la figura 3.5. se puede observar el diagrama de casos de uso del Administrador, este puede subir y publicar una imagen para ser renderizada, así como eliminarla, también puede descargar las imágenes, administrar las mismas y subirlas al servidor.

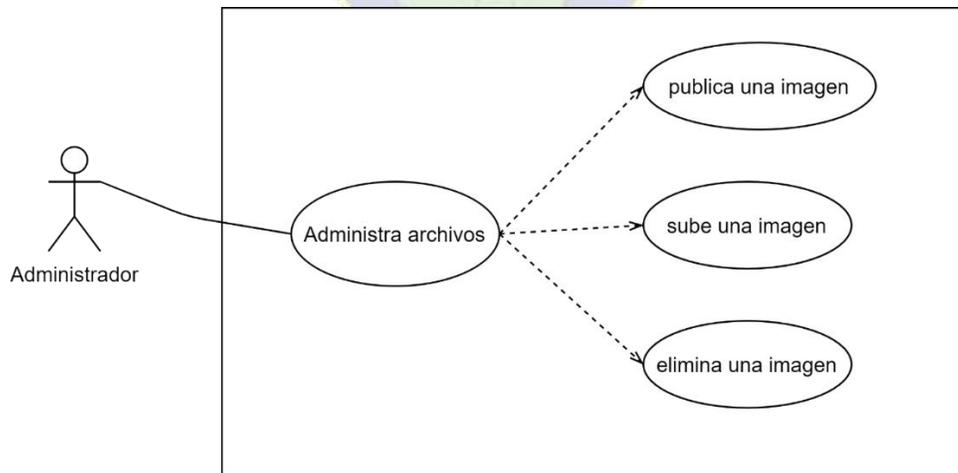


Figura 3.5. Diagrama de casos de uso – Administrador

Fuente: elaboración propia

- Desarrollo: en la tabla 3.3 se detalla el desarrollo del primer sprint backlog que tiene como enfoque la conexión a servicios en la nube y su primera implementación a la aplicación web a elaborar.

Tabla 3.3. Sprint 1 – Servicios en la nube

SPRINT N°1 – SERVICIOS EN LA NUBE		
ID	PROCESO	SUBPROCESO
1	Diseño de una conexión a servicios en la nube.	Creación de cliente en la nube.
		Manejo de usuarios en la nube.
		Manejo de archivos en la nube.
2	Administración de archivos dentro de la aplicación.	Organización de archivos en la nube.
		Conexión con la aplicación.
		Conexión con archivos de la aplicación web.

Fuente: Elaboración propia

En la tabla 3.3 se observa el proceso que se realiza para habilitar la conexión, en la siguiente tabla se construyen las historias de usuario, tabla 3.4. En la tabla 3.4. se contempla la conexión a servicios en la nube.

Tabla 3.4. Historias de usuario – Conexión de servicios en la nube

HISTORIAS DE USUARIO	
Número: 1	Usuario: Desarrollador
Nombre de la historia: Diseño de la conexión a servicios en la nube.	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 8	Interacción asignada: 1
Programador responsable: Alberth Bernarth Condori Quispe	
<p>Descripción: El módulo se enfoca en el diseño de la conexión a servicios en la nube, esta contempla los siguientes puntos:</p> <ul style="list-style-type: none"> • Conexión a servicios en la nube con Amazon Web Services. • Crear la conexión con la aplicación web. • Crear el test de envío y recepción de archivos. 	

Fuente: Elaboración propia

En la tabla 3.5 se observa la historia de usuario correspondiente a el Administrador y las funciones que contempla en esta sección.

Tabla 3.5. Historias de usuario – Funciones para Administrador

HISTORIAS DE USUARIO	
Número: 2	Usuario: Administrador
Nombre de la historia: Administración de archivos.	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 5	Interacción asignada: 1
Programador responsable: Alberth Bernarth Condori Quispe	
<p>Descripción: El módulo se enfoca en el diseño de los módulos y modelos necesarios para permitir al Administrador el manejo de archivos contemplando los siguientes puntos:</p> <ul style="list-style-type: none"> • Publicar una imagen. • Subir una imagen. • Borrar una imagen <p>Además con el framework de Django el Administrador podrá también visualizar todos los archivos subidos a la aplicación web, de forma detallada.</p>	

Fuente: Elaboración propia

- Revisión: en esta etapa del primer sprint se verificó el cumplimiento de las tareas planificadas, lo que significa que estas tareas se concluyeron de manera satisfactoria.

La tabla 3.6 es la prueba de historia de usuario que muestra pruebas con casos de uso sobre la primera iteración.

Tabla 3.6. Prueba de historias de usuario – Servicios en la nube

PRUEBA DE HISTORIAS DE USUARIO	
Caso de prueba: 1	Número de historia de usuario: 1 y 2
Descripción de la prueba: Prueba para servicios en la nube utilizando archivos estáticos e imágenes, para el propósito de la aplicación web.	
Ejecución: Amazon Web Services recibe una petición desde el framework, con todos los permisos necesarios, y mediante el administrador del mismo se realiza la carga del archivo, una vez finalizado este registro aparece en la base de datos y a su vez la imagen se carga en la nube.	
Entrada: La entrada son imágenes o archivos estáticos que serán cargados en Amazon Web Services.	
Resultado esperado: Las imágenes son desplegadas en el navegador, por el administrador que ofrece el framework Django y también están disponibles en el árbol de archivos en Amazon Web Services.	

Fuente: Elaboración propia

A continuación, se observa, en la figura 3.6 la configuración de la cubeta de información que contendrá todos los archivos estáticos y de multimedia de la aplicación web. Se debe notar que la región seleccionada en la configuración de servicios en la nube está situada en el Norte de Virginia, esta configuración es la predeterminada para utilizar el servicio sin coste. Sin embargo, esto significa que la cubeta tiene un límite, si dicho límite es superado se borrarán los archivos más antiguos.

Configuración general

Nombre del bucket

Región

Figura 3.6. Configuración inicial para el alojamiento de archivos

Fuente: Elaboración propia

Cuando la cubeta de información es creada es necesario también crear un ID de clave de acceso para la misma. En la figura 3.7. es posible observar la clave de acceso a Amazon Web Services y en la figura 3.8. se puede apreciar el uso de la misma para la aplicación web en Django, además del nombre de la cubeta previamente configurado que se requiere para correcta conexión con los servicios en la nube.

ID de clave de acceso	Creado	Estado
AKIAV3VJIJD3DRGJFSQT	██████████	Activa Desactivar

Figura 3.7. Clave de acceso a una cuenta en Amazon Web Services

Fuente: Elaboración propia

```
1 AWS_ACCESS_KEY_ID = 'AKIAV3VJIJD3DRGJFSQT'
2 AWS_SECRET_ACCESS_KEY = '
3 AWS_STORAGE_BUCKET_NAME = 'cubeta-de-datos'
```

Figura 3.8. Configuración de claves de acceso para Amazon Web Services

Fuente: Elaboración propia

En la figura 3.9. se puede observar los permisos otorgados a la aplicación web que accede a los servicios en la nube provistos por Amazon Web Services. Estos permisos incluyen: bajar, subir y borrar archivos.

Uso compartido de recursos entre orígenes (CORS)

```
1 - [
2 - {
3 -   "AllowedHeaders": [
4 -     "*"
5 -   ],
6 -   "AllowedMethods": [
7 -     "PUT",
8 -     "POST",
9 -     "DELETE"
10 -  ],
11 -   "AllowedOrigins": [
12 -     "*"
13 -   ],
14 -   "ExposeHeaders": []
15 - }
16 - ]
```

Figura 3.9. Permisos de acceso a los archivos

Fuente: Elaboración propia

Para el alojamiento de archivos en la nube se organizan los archivos en carpetas etiquetadas con diferentes nombres como se puede observar en la figura 3.10.

cubeta-de-datos

Objetos

Objetos (5)

Nombre ▲	Tipo ▼	Última modificación
css/	Carpeta	-
draws/	Carpeta	-
images/	Carpeta	-
img/	Carpeta	-
pdf/	Carpeta	-

Figura 3.10. Organización de carpetas y objetos en la nube

Fuente: Elaboración propia

De esta forma se obtiene una conexión con Amazon Web Services y la posibilidad de administrar los archivos desde la aplicación web bajo el framework Django.

3.4.2.2. SPRINT 2 ADMINISTRACIÓN CON FORMULARIOS

- **Planeación:** Para la administración con funciones básicas desde la aplicación web es necesario crear formularios, que se comuniquen directamente con el almacenamiento de archivos en la nube. De esta manera también hacemos posible que desde el mismo Framework de Django podamos procesar imágenes y consultar a la API.

En la siguiente figura 3.11 de casos de uso se representan los procedimientos necesarios que el desarrollador sigue para el diseño de estos formularios.

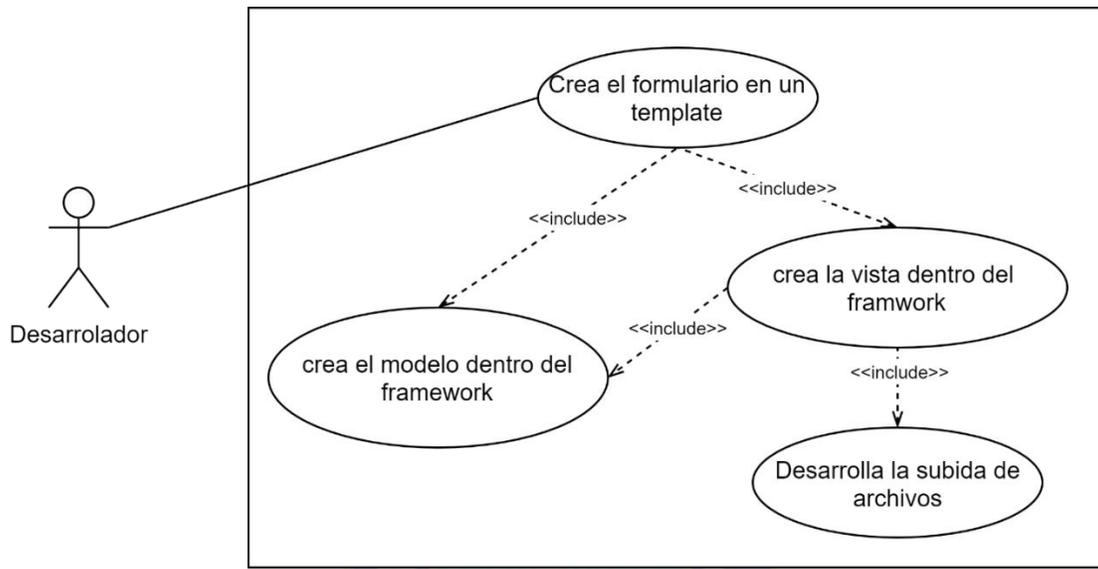


Figura 3.11. Diagrama de casos de uso, creación de formularios – Desarrollador

Fuente: Elaboración propia

En la siguiente figura 3.12 de casos de uso se presentan los procedimientos de los cuales es actor el administrador. Estos procedimientos son importantes a la hora de crear y publicar nuevos dibujos en la aplicación web. Esta tarea es primordial, por estar relacionada con el objetivo principal del producto final y la misma será llevada a cabo por el administrador.

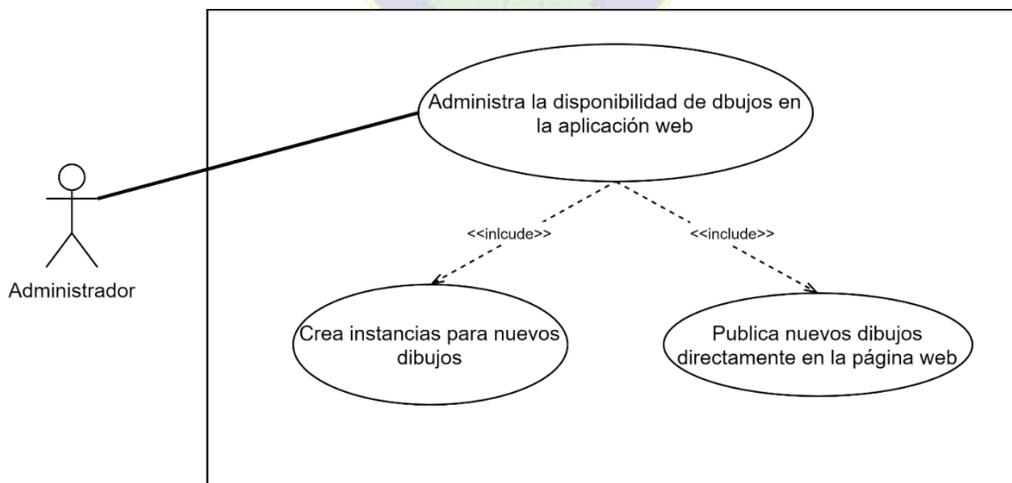


Figura 3.12. Diagrama de casos de uso, creación de formularios – Administrador

Fuente: Elaboración propia

- Desarrollo: En la tabla 3.7 se detalla el desarrollo del segundo sprint backlog que tiene como enfoque la administración con formularios de las imágenes disponibles para dibujar, así como también de la interacción del usuario simple con la API.

Tabla 3.7. Sprint 2 – Administración con formularios

SPRINT N°2 – MANEJO DE FORMULARIOS		
ID	PROCESO	SUBPROCESO
3	Manejo de formularios para administrar archivos.	Creación de formulario para la API.
		Creación de formulario para los dibujos.
		Manejo de archivos con formularios.

Fuente: Elaboración propia

En la tabla 3.8 se observa el proceso que se realiza la creación o estructuración de formularios.

Tabla 3.8. Historias de usuario – Creación de formularios

HISTORIAS DE USUARIO	
Número: 3	Usuario: Desarrollador
Nombre de la historia: Creación de formularios.	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 8	Interacción asignada: 1
Programador responsable: Alberth Bernarth Condori Quispe	
Descripción: El módulo se enfoca en el diseño y la construcción de formularios que hagan posible la administración de los archivos almacenados en la nube. Este módulo contempla los	

siguientes puntos:

- Creación de formulario para la API.
- Construcción de formularios para la carga de imágenes nuevas.
- Creación de formularios de acceso a las imágenes cargadas.

Fuente: Elaboración propia

En la tabla 3.9 en la siguiente tabla se observa la administración de dibujos por parte del usuario de tipo administrador.

Tabla 3.9. Historias de usuario – Manejo de formularios

HISTORIAS DE USUARIO	
Número: 4	Usuario: Administrador
Nombre de la historia: Creación de formularios.	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 5	Interacción asignada: 1
Programador responsable: Alberth Bernarth Condori Quispe	
Descripción: El siguiente módulo se enfoca en el desarrollo de formularios necesarios para la administración de imágenes disponibles para ser dibujos.	

Fuente: Elaboración propia

- Revisión: En esta etapa del segundo sprint se verificó el cumplimiento de las tareas planificadas, lo que significa que estas tareas se concluyeron de manera satisfactoria.

La tabla 3.10 es la prueba de historia de usuario que muestra pruebas con casos de uso sobre la segunda iteración.

Tabla 3.10. Prueba de historias de usuario. Servicios en la nube

PRUEBA DE HISTORIAS DE USUARIO	
Caso de prueba: 2	Número de historia de usuario: 3 y 4
Descripción de la prueba: Prueba para formularios utilizando archivos desde origen local, para posterior interacción con una API y realidad aumentada.	
Ejecución: El formulario es la interfaz para recibir la imagen cargada. Una vez que la imagen es cargada se puede proceder al procesamiento o el alojamiento de las misma. Otro formulario procede a cargar las imágenes que el administrador subió para su visualización en otra página.	
Entrada: La entrada son imágenes cargadas desde un origen local.	
Resultado esperado: Las imágenes son desplegadas con o sin respuesta por el framework dependiendo al formulario al que se cargó la imagen.	

Fuente: Elaboración propia

Para guardar las imágenes con los formularios provistos para archivos, en este caso imágenes, es necesario crear formularios desde el apartado de modelos de Django (Figura 3.13.) y al momento de guardar también se debe analizar estas imágenes y deben ser procesadas antes de ser subidas a la nube, en ambos casos tanto como para las imágenes con propósito de ser los bocetos mostrados por la realidad aumentada, como los subidos (Figura 3.14.). De este modo cuando se requiera comparar un dibujo con el original se tendrá la imagen lista para ser procesada por la API.

Para procesar la imagen se requiere de convertir la imagen a blanco y negro con OpenCV (Figura 3.15.), posteriormente realizamos un proceso de umbralización que es obtener solo los trazos de las imágenes, es decir, obtener todas las partes de la imagen que contenga todos los negros representativos de la imagen logrando así quitar el ruido que pueda existir en la foto y finalmente reconocer solo las líneas para que sean analizadas con la API como se puede observar en la Figura 3.16.

```
1 class Images(models.Model):
2     image_dibujo = models.ImageField(upload_to='images')
3
4     def __str__(self):
5         return str(self.pk)
```

Figura 3.13. Modelo Django para imágenes

Fuente: Elaboración propia

```
1 class Images(models.Model):
2     image_dibujo = models.ImageField(upload_to='images')
3
4     def __str__(self):
5         return str(self.pk)
```

Figura 3.14. Modelo de formularios para imágenes

Fuente: Elaboración propia

```
1 gImg = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

Figura 3.15. Imagen a blanco y negro con OpenCV

Fuente: Elaboración propia

```
1 cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY,11,3)
```

Figura 3.16. Obtener partes representativas de la imagen

Fuente: Elaboración propia

3.4.2.3. SPRINT 3 MANEJO DE UNA API

- Planeación: Para el manejo de una API solo se requiere interactuar con esta, mediante código. Para este procedimiento lo único que se requiere es llamar a esta API, para esto es necesario interactuar con la vista de la aplicación web, provista por el framework Django.

En la siguiente figura 3.17 de casos de uso se representan los procedimientos necesarios que el desarrollador sigue para el diseño de estos formularios.

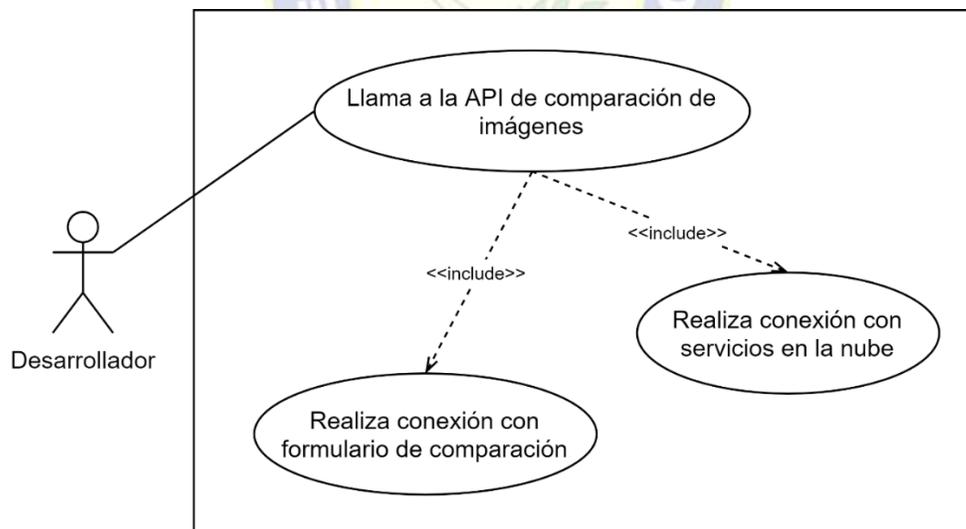


Figura 3.17. Diagrama de casos de uso, manejo de API – Desarrollador

Fuente: Elaboración propia

El usuario simple es el actor más importante para la aplicación web, pues es este quién realizará y tendrá contacto con las funciones principales de la aplicación web. Estas

funciones para este sprint son: subir imágenes y obtener un resultado enviado por la API sobre la comparación de imágenes procesada. En la figura 3.18 se puede observar su diagrama de casos de uso.

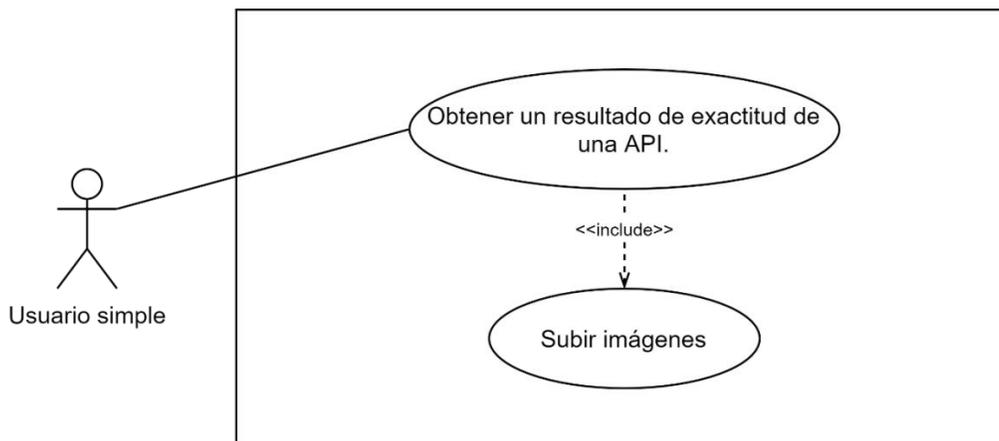


Figura 3.18. Diagrama de casos de uso, manejo de API – Usuario simple

Fuente: Elaboración propia

- Desarrollo: En la tabla 3.11 se detalla el desarrollo del tercer sprint backlog que tiene como enfoque el manejo de la API con el Framework Django y bajo el lenguaje de programación Python.

Tabla 3.11. Sprint 3 – Manejo de API

SPRINT N°3 – MANEJO DE API		
ID	PROCESO	SUBPROCESO
4	Acciones para el manejo de una API en la aplicación web	Uso de la API con sus parámetros.
		Conexión con formularios.
		Conexión con servicios en la nube.

Fuente: Elaboración propia

En la tabla 3.12 se describen los procesos que requiere el manejo de API para la recepción de archivos estáticos, los cuales en este caso son imágenes que serán subidas y analizadas por la API utilizada desde el framework Django.

Tabla 3.12. Historias de usuario – Conexión con API

HISTORIAS DE USUARIO	
Número: 5	Usuario: Desarrollador
Nombre de la historia: Conexión con API	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 8	Interacción asignada: 1
Programador responsable: Alberth Bernarth Condori Quispe	
<p>Descripción: El módulo se enfoca en la conexión de la API con los servicios y módulos correctos para el buen funcionamiento y de esta manera obtener respuesta de la API a utilizar. Para esto se realizan las siguientes tareas:</p> <ul style="list-style-type: none"> • Llamada a la API desde el framework Django. • Conexión de la API con el formulario correspondiente. • Conexión de la API con los servicios de almacenamiento en la nube. 	

Fuente: Elaboración propia

En la tabla 3.13 se observa la historia de usuario perteneciente al usuario simple. Es aquí donde se desarrollan sus funciones en interacción con los servicios en la nube.

Para esta parte se debe tomar en cuenta que solo se toma en cuenta la carga de imágenes, siendo el actor usuario simple quién realiza esta tarea al tener interacción con los servicios en la nube.

Tabla 3.13. Historias de usuario – Funciones para Usuario Simple

HISTORIAS DE USUARIO	
Número: 6	Usuario: Usuario simple
Nombre de la historia: Carga de imágenes y obtención de respuesta	
Prioridad: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 5	Interacción asignada: 1
Programador responsable: Alberth Bernarth Condori Quispe	
Descripción: Analizar, diseñar y crear las funciones necesarias para que el usuario sea capaz de subir imágenes para así utilizar una API y obtener respuesta.	

Fuente: Elaboración propia

- **Revisión:** En esta etapa del tercer sprint se verificó el cumplimiento de las tareas planificadas, lo que significa que estas tareas se concluyeron de manera satisfactoria.

La tabla 3.14 es la prueba de las historias de usuario que muestra pruebas con casos de uso sobre la tercera iteración.

Tabla 3.14. Prueba de historias de usuario. Servicios en la nube

PRUEBA DE HISTORIAS DE USUARIO	
Caso de prueba: 3	Número de historia de usuario: 5 y 6
Descripción de la prueba: Prueba de la conexión y uso de la API utilizando imágenes.	
Ejecución: La API es capaz de interactuar a través del framework con los formularios y este a su vez es capaz de interactuar con los servicios en la nube, de esta forma se realiza la carga de una imagen y es analizado mediante el alojamiento de archivos provistos por los servicios en la nube.	
Entrada: Imágenes estáticas desde el alojamiento en la nube.	
Resultado esperado: Respuesta numérica de la API.	

Fuente: Elaboración propia

Cabe resaltar que para llamar a la API empleada desde Python tenemos dos formas con un response simple (Figura 3.19.) o con procesos asíncronos llamando a un método desde la librería que provee la API (Figura 3.20.)

Como comprobación sobre el proceso realizado al analizar las imágenes revisar el capítulo 3, medida del índice de similitud estructural, del presente trabajo, el cual esta implementado en la aplicación web. (Figura 3.20.)

```
1 r = requests.post(
2     "https://api.deepai.org/api/image-similarity",
3     data={
4
5         'image1': str(dibujo.image.url),
6         'image2': str(img_obj.image_dibujo.url),
7
8     },
9     headers={'api-key': '48c310e7-5fba-4ff5-8483-67281b2a0e78'})
10
11 ans = str(r.json()['output']['distance'])
```

Figura 3.19. Obtención de resultado por response.

Fuente: Elaboración propia

```
1 respuesta = await sync_to_async(see_similarity, thread_sensitive=True)(img1, img2)
```

Figura 3.20. Obtención de resultado por un método

Fuente: Elaboración propia

3.4.2.4. SPRINT 4 INTEGRACIÓN DE REALIDAD AUMENTADA

- Planeación: Para la integración de realidad aumentada se empleará también la metodología a usar, propuesta previamente en este capítulo. Para esto es necesario interactuar con el almacenamiento en la nube.

En la siguiente figura 3.21 se muestra los casos de uso necesarios para que el desarrollador tenga las herramientas para desarrollar e integrar realidad aumentada a la aplicación web.

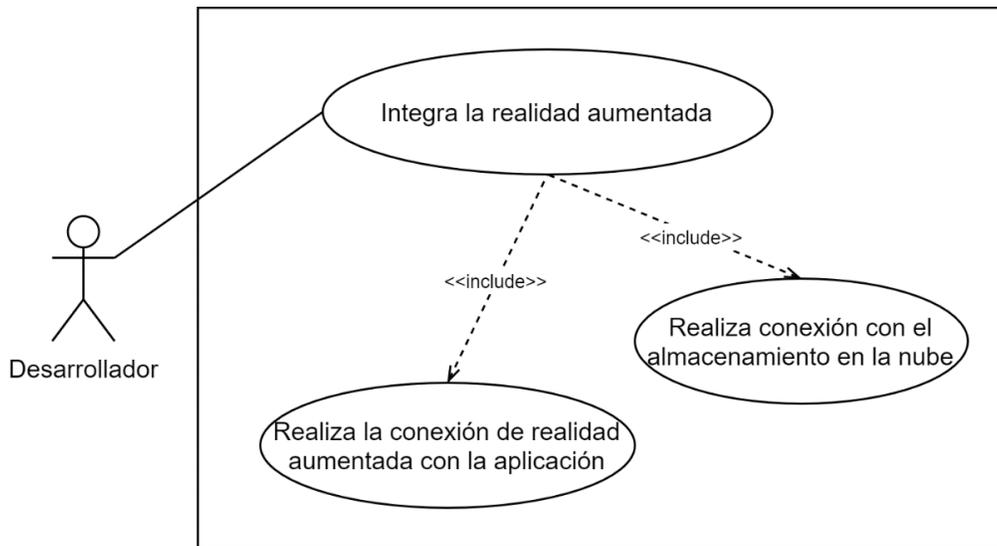


Figura 3.21. Diagrama de casos de uso, integración de realidad aumentada

Fuente: Elaboración propia

El usuario simple es el actor principal en el empleo de este sprint, pues es este quién interactuará de cerca con las funcionalidades que ofrece la realidad aumentada integrada en esta aplicación web. Y es este quién hace una elección sobre el dibujo que dibujará y por lo tanto es quién escoge un boceto para que sea renderizado por cámara a través de la aplicación web. Figura 3.22.

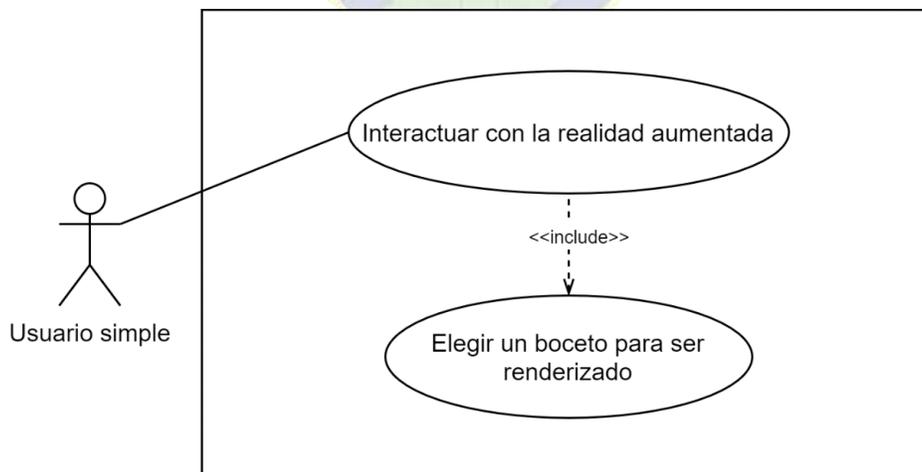


Figura 3.22. Diagrama de caso de uso, interacción con realidad aumentada

Fuente: Elaboración propia

- Desarrollo: para la parte de desarrollo, y la parte de revisión más adelante (nombrada ahora como evaluación para la concordancia con la metodología), se empleará la metodología propuesta para el desarrollo de realidad aumentada.

Diseño: Para la parte de diseño se elige o produce los elementos a usar para el trabajo con realidad aumentada. Para esto requerimos usar elementos para mostrar y elementos para reconocer, para este trabajo estos son los elementos identificados.

El elemento para reconocer en este caso es un marcador, por su sencillez frente a una imagen u objeto, este marcador es en blanco y negro y con pocos detalles para que la calidad de la cámara a emplear no sea un factor primordial a la hora de realizar las pruebas. Figura 3.24.

Los elementos para mostrar son todos de carácter artístico, por lo tanto, todos son bocetos, para el propósito de este trabajo. Un punto a tomar en cuenta sobre los elementos a mostrar es que estos no son fijos y pueden ser borrados, cambiados o hasta actualizados. Incrementando o disminuyendo así su cantidad. Sin embargo, los elementos a mostrar para este trabajo serán tomados de un manual de dibujo. Figura 3.23.

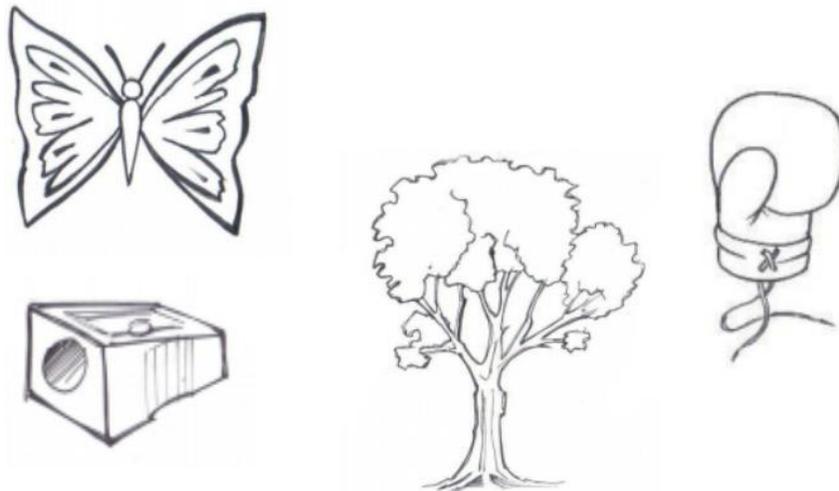


Figura 3.23. Elementos para mostrar

Fuente: (Ramonfaur, 2018)

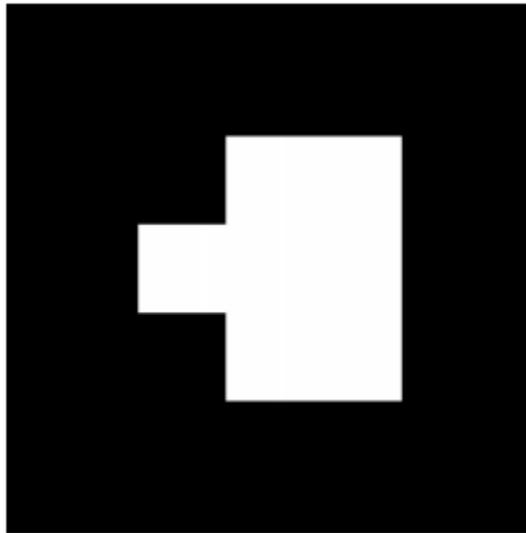


Figura 3.24. Elemento para reconocer

Fuente: Elaboración propia

Implementación: Para implementar estos elementos en la realidad aumentada e integrarlos se depende de la obtención de estos elementos. Para la obtención de estos elementos recurriremos a el almacenamiento en la nube y no así a un fichero estático. Sin embargo, para el elemento que tenemos que reconocer optaremos, para el propósito de esta aplicación con la impresión del mismo.

Para mostrar un elemento recurrimos al framework Django y mediante código consultamos a nuestra base de datos almacenada en una cubeta en almacenamiento en la nube.

```
<a-assets>
  
</a-assets>
```

Figura 3.25. Implementación de elementos para mostrar

Fuente: Elaboración propia

Adicionalmente, se debe tomar en consideración que para renderizar la imagen debemos hacer que esta tenga un tamaño aproximadamente igual para todos los bocetos. Para esta tarea tomaremos en consideración la proporción de la imagen. (Figura 3.26.)

```
1  realWidth = my_imagen.naturalWidth;
2  realHeight = my_imagen.naturalHeight;
3
4  let ratio = realWidth / realHeight;
5
6  if (ratio < 1) {
7    newHeight = (1/ratio).toFixed(2);
8    newWidth = 1;
9  } else {
10   newHeight = (1.8/ratio).toFixed(2);
11   newWidth = 1.8;
12 }
```

Figura 3.26. Reestablecer el tamaño de la imagen

Fuente: Elaboración propia

Evaluación: Para la evaluación se requiere que la integración de las demás tecnologías, a implementar, estén correctamente integradas a la aplicación web. Como el reconocimiento del marcador está sujeto a la calidad de la cámara a emplear, es necesario crear una estabilidad entre la imagen a reconocer y la imagen a mostrar. Al final obtenemos el siguiente resultado preliminar (Figura 3.27.)

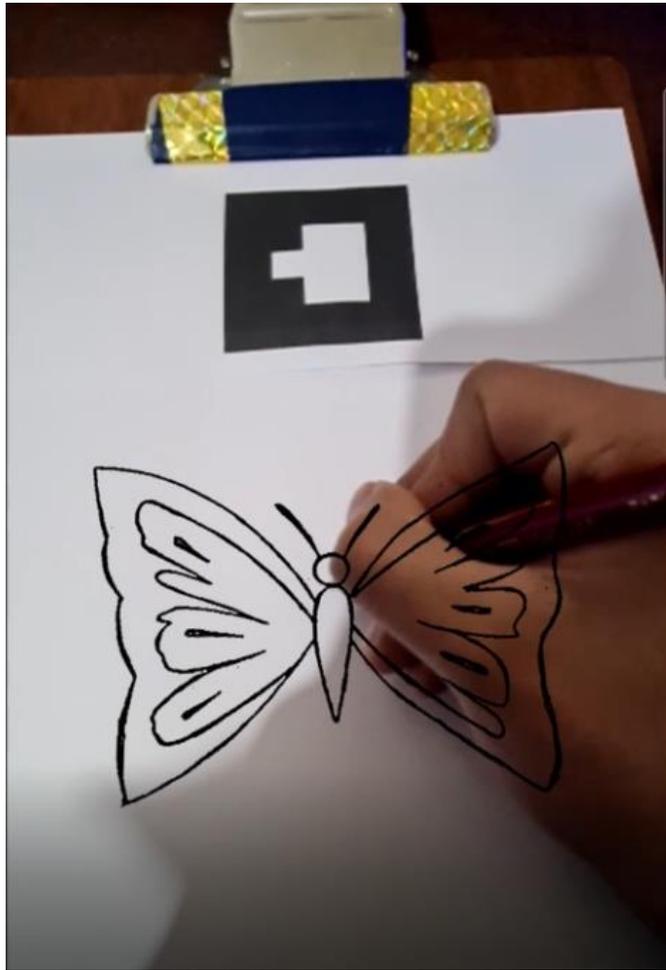


Figura 3.27. Muestra de la vista con realidad aumentada

Fuente: Elaboración propia



CAPÍTULO 4 PRUEBA DE HIPÓTESIS

4.1. INTRODUCCIÓN

En el presente capítulo se realiza la prueba de verificación de la hipótesis planteada, con el uso de la misma y la recolección de datos mostrados por la API, mostrando un porcentaje de semejanza. Por lo que sabremos que si se ha logrado hacer un boceto con la aplicación.

A continuación, se establece la hipótesis, planteada en el capítulo uno, y la contra hipótesis que nos ayudará a demostrar la validez de la hipótesis principal.

4.2. ANÁLISIS

En esta sección se realiza el análisis según la metodología de investigación que se sigue en el presente trabajo como una etapa del método científico.

Se definen la hipótesis nula y la hipótesis alternativa:

H_0 : El uso de la realidad aumentada en conjunto con una API de inteligencia artificial, hace posible dibujar bocetos sencillos con líneas, semejantes al modelo original con un error del 10%.

H_1 : El uso de la realidad aumentada en conjunto con una API de inteligencia artificial, no hace posible dibujar bocetos sencillos con líneas, semejantes al modelo original con un error del 10%.

4.2. EXPERIMENTACIÓN

Se realiza la experimentación determinando las siguientes actividades:

- Determinación de la población
- Determinación de la muestra
- Proceso de experimentación

- Análisis de los resultados

De la hipótesis se han planteado la variable dependiente como la variable independiente en el capítulo 1.

- Variable independiente:

Realidad aumentada en conjunto con la API de inteligencia artificial.

- Variable dependiente:

Dibujo de bocetos sencillos con líneas.

Es importante mencionar que son personas las que realizarán los bocetos y son sus bocetos relacionados con su puntaje de semejanza lo que será estudiado.

- a) Determinación de la población:

La población a tomar en cuenta son los bocetos realizados por usuarios simples. Específicamente se tomarán los resultados de semejanza arrojados por la API de inteligencia artificial. Se tomarán muestras de 3 dibujos elegidos como propios para esta prueba tomados del manual de dibujo (Ramonfaur, 2018) Dibujo artístico – Nociones básicas, que nos ofrece una variedad de dibujos útiles y recomendados para trabajar con bocetos, se puede ver estos tres dibujos en las figuras a continuación:



Figura 4.1. Boceto número 1

Fuente: Elaboración propia

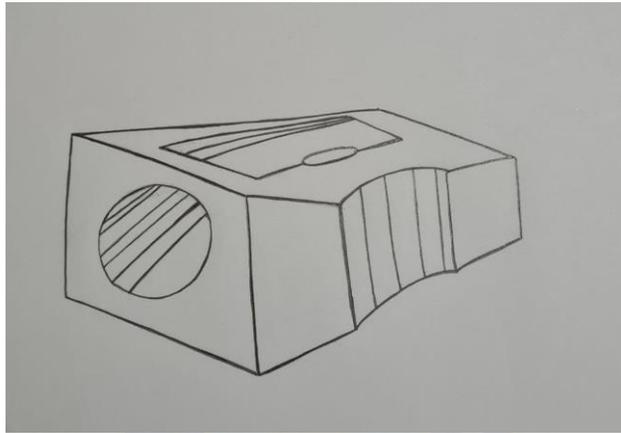


Figura 4.2. Boceto número 2

Fuente: Elaboración propia

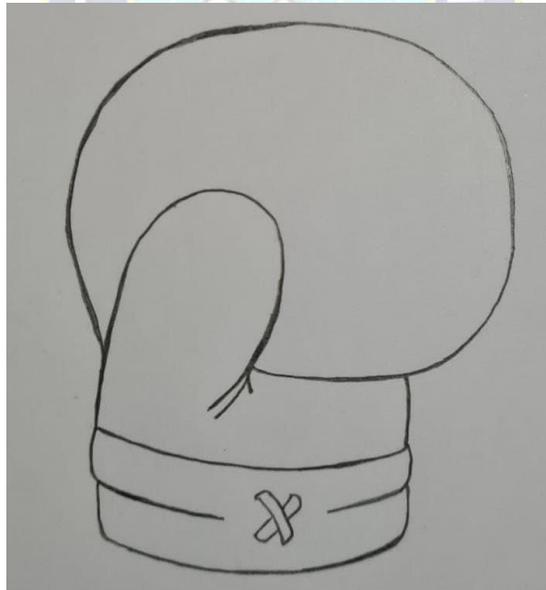


Figura 4.3. Boceto número 3

Fuente: Elaboración propia

b) Determinación de la muestra:

Se toman como muestra 5 dibujos realizados de cada uno de los 3 dibujos disponibles en la aplicación web. Por lo tanto, la cantidad total de dibujos que se toman en cuenta son 15 dibujos realizados por diferentes usuarios simples.

Para el primer dibujo se obtuvieron muestras con el boceto 1 Figura 4.1. Tres de estas muestras son las siguientes: (Figura 4.4)

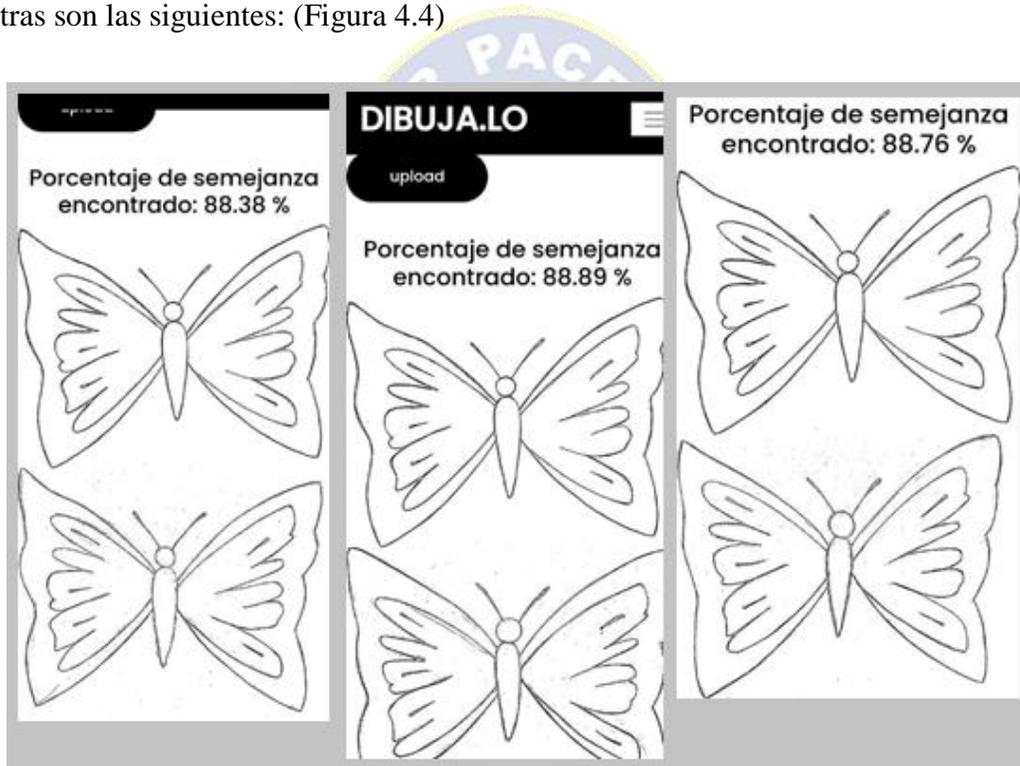


Figura 4.4. Resultados con el primer boceto

Fuente: Elaboración propia

Para el primer dibujo se obtuvieron muestras con el boceto 2. Figura 4.2. Tres de estas muestras son las siguientes: (Figura 4.5)

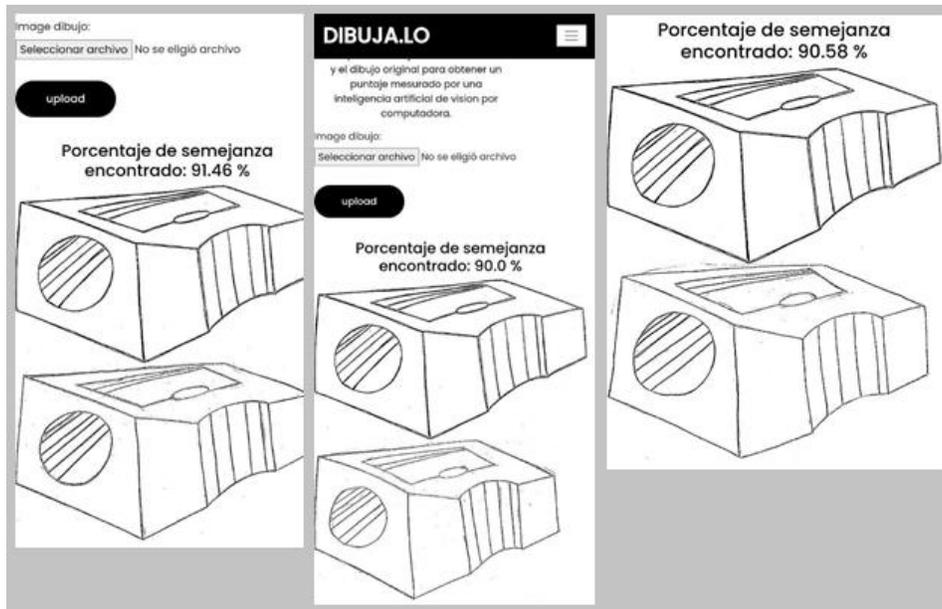


Figura 4.5. Resultados con el segundo boceto

Fuente: Elaboración propia

Para el primer dibujo se obtuvieron muestras con el boceto 3. Figura 4.3. Tres de estas muestras son las siguientes: (Figura 4.6)

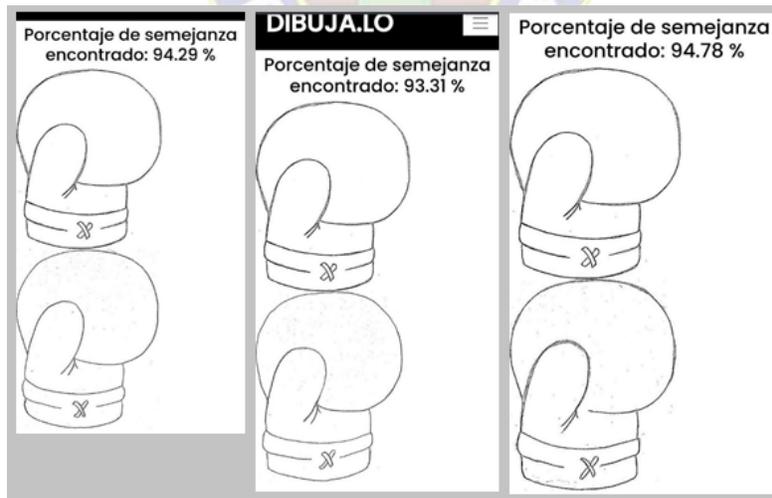


Figura 4.6. Resultado con el tercer boceto

Fuente: Elaboración propia

c) Proceso de experimentación:

Primero se recolecta los resultados lanzados por la API de inteligencia artificial por cada dibujo realizado es así que se obtiene como resultado la siguiente información:

Tabla 4.1. Puntajes tomados de la API

N	Dibujo 1	Dibujo 2	Dibujo 3	Promedio
1	84,38	91,46	94,29	90,0433
2	88,89	90	93,31	90,7333
3	88,76	90,58	94,78	91,3733
4	87,33	90,24	92,48	90,0167
5	85,81	89,83	91,57	89,0700

Siendo así el promedio de los resultados.

$$\bar{x} = \frac{451,2366}{5} = 90,247$$

Ecuación 4.1. Media muestral

d) Análisis de los resultados:

Tomando de la hipótesis nula

$$H_0: P = 90$$

$$H_1: P < 90$$

El estadístico de contraste en este caso es para una sola muestra para debido a las características de los datos recolectados: (Tabla 4.1)

$$t = \frac{\bar{x} - \mu_0}{\frac{\sigma}{\sqrt{n-1}}}$$

Ecuación 4.2. Estadístico de contraste

Se requiere para el estadístico de contraste la media muestral y la desviación típica muestral, la cual es descrita a continuación:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} = 0,864$$

Ecuación 4.3. Desviación típica muestral

Substituyendo en el estadístico y con 4 grados de libertad se obtiene:

$$t = \frac{90,247 - 90}{\frac{0,864}{\sqrt{4}}} = 0,5718$$

Ecuación 4.4. Resultado estadístico de contraste

Obtenemos con la tabla t-Student el resultado de 1,5332, el valor obtenido nos indica que rechazamos la hipótesis alternativa, esto significa que aceptamos nuestra hipótesis nula, debido a que el valor obtenido está dentro de la región de aceptación.

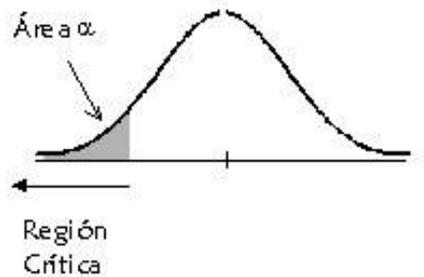


Figura 4.7. Región de aceptación

Fuente: (Universidad de Granada, 2021)

4.3. ANÁLISIS

Los resultados finales de la etapa de experimentación con los datos de semejanza obtenidos muestran que se logró la demostración de la hipótesis al 10% de error que es lo que se espera cuando se depende de una evaluación exhaustiva tomando en cuenta la semejanza total de dos imágenes obteniendo una semejanza esperada a lo que debe ser respecto al boceto original.



CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1. INTRODUCCIÓN

A continuación, en los siguientes puntos se describen los resultados tras obtener resultados en la prueba de hipótesis, logrando así la prueba del mismo. Así como recomendaciones a tomar en cuenta a la hora de hacer uso de la aplicación web.

5.2. CONCLUSIONES

Ya culminada la aplicación web y hecha sus pruebas según las metodologías empleadas para el propósito de este trabajo se concluye el cumplimiento de los objetivos planteados en el capítulo 1.

- Se logró implementar realidad aumentada a la aplicación web con el objetivo de visualizar dibujos que pueden ser cargados desde la misma página web, evitando así tener que cargarlos en producción.
- Se implementó con éxito una API de inteligencia artificial que logra obtener un resultado que describe la semejanza de dos imágenes, que para el objetivo de este trabajo son bocetos hechos en papel a los cuales se les tomo fotografías, así también cabe mencionar que estas imágenes fueron tratadas y limpiadas para analizar solo los trazos hechos.
- Se logró también implementar a la aplicación web almacenamiento en la nube para guardar los bocetos a dibujar. Así como los bocetos tomados desde cámara para ser comparados por la API.
- También se logró integrar las tecnologías necesarias para la muestra de diferentes bocetos a través de la realidad aumentada.
- Se creó una interfaz que integra dentro de la aplicación web, todas las tecnologías mencionadas dentro del trabajo para obtener resultados por la API.

5.3. RECOMENDACIONES

La elaboración del proyecto depende de varios tipos de tecnologías, pero al ser implementadas dentro de un framework es posible separarlas y que trabajen de manera conjunta. Sin embargo, no es posible separar una tecnología para que las demás trabajen independientemente, todas deben estar bajo correcto funcionamiento. Lo que supone una constante revisión de las tecnologías implementadas en la aplicación web.

Para la implementación y correcto funcionamiento de la realidad aumentada en conjunto con el almacenamiento en la nube fue necesario crear políticas de acceso a las imágenes que se renderizan desde el almacenamiento en la nube, mencionadas en el capítulo 3, por lo que no se puede hacer más de 20 dibujos por hora. Este hecho es necesario porque para el propósito de este trabajo todo debe ser de carácter libre.

A pesar de que todo es libre se restringe el acceso directo a los archivos almacenados en la nube, debido a que estos están sujetos a la aplicación web, la modificación de los mismos depende del administrador o administradores de la aplicación web, como se describió en el capítulo 3.

Esta aplicación depende mucho de la calidad de cámara que se vaya a emplear tanto para la realidad aumentada, así como la calidad de fotografías que se sube a la aplicación para que sean comparadas con el boceto original.

Otra aplicación que puede tener la parte de reconocimiento de líneas, del presente trabajo, puede aplicarse a el reconocimiento de dibujos mejor elaborados al quitarles el fondo recortarlos y posteriormente utilizar los mismos con diferentes propósitos.

Bibliografía

Alcarria Izquierdo Carlos (2010), Desarrollo de un Sistema de Realidad Aumentada en dispositivos móviles [en línea] <https://www.academia.edu/36366396/UNIVERSIDAD_POLIT%C3%89CNICA_DE_VALENCIA_ESCUELA_T%C3%89CNICA_SUPERIOR_DE_INGENIER%C3%89DA_INFORM%C3%89TICA_PROYECTO_FINAL_DE_CARRERA?email_work_card=view-paper> [consulta: 15 de Septiembre de 2020].

Alvalyn Lundgren (2019). 8 Consejos para mejorar en el dibujo [en línea] <<https://alvalyn.com/8-tips-for-improving-your-drawing-skills/>> [consulta: 28 de Julio de 2020].

Alvarado Yoselie (2019), Aplicaciones de Realidad Virtual y Realidad Aumentada como soporte a la enseñanza del Dibujo Técnico [en línea] <<https://digibug.ugr.es/handle/10481/58153>> [consulta: 29 de Julio de 2020].

Ariel, Fundación Telefónica (2015). Qué es la realidad aumentada. (Primera edición). Realidad aumentada: una nueva mente para ver el mundo (págs: 9-98). España. [en línea] <<https://books.google.com.bo/books?id=OXHmCgAAQBAJ&lpg=PA10&ots=3rv7X Xdmta&dq=realidad%20aumentada&lr=&hl=es&pg=PA10#v=onepage&q=realidad%20aumentada&f=false>> [consulta 7 de Septiembre de 2020].

Avilés Olmos Marina, Mirete Ruiz Ana Belen, Maquilon Sanchez (2016), Innovación educativa basada en realidad aumentada: Una experiencia en el área de dibujo. [en línea] <<https://digitum.um.es/digitum/handle/10201/87450>> [consulta: 29 de Julio de 2020].

Boden Margaret (2017), Inteligencia Artificial [en línea] <<https://books.google.com.bo/books?hl=es&lr=&id=LCnYDwAAQBAJ&oi=fnd&pg=PT3&dq=inteligencia+artificial&ots=drPIBXaLjd&sig=la003OIsd->>

bRDB4g2AjwPH1ZXMw&redir_esc=y#v=onepage&q=inteligencia%20artificial&f=fa
alse> [consulta: 20 de Septiembre de 2020].

Deshpande Adit (2016), A Beginner's Guide To Understanding Convolutional Neural Networks [en línea] <<https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>> [consulta: 25 de Septiembre de 2020].

Diccionarios Everest (2015). Diccionario de la lengua española (11 ed.). Madrid, Barcelona.

Drobitko Andrey (2021). SketchAR [en línea] <<https://sketchar.io/>> [consulta: 20 de Mayo de 2021].

Fernandez Carlos Enrique (2020). Tócame que soy realidad aumentada <<http://realidadaumentadaperu.blogspot.com/>> [consulta: 10 de Julio de 2020].

Hernández Tubío Ángel Luis (2016), Realidad aumentada en dibujo técnico [en línea] <<http://tauja.ujaen.es/handle/10953.1/2675>> [consulta: 29 de Julio de 2020].

Jaime Rubio Hancock (2015), Los beneficios de dibujar y colorear de adulto, aunque no tengas ni idea [en línea] <https://verne.elpais.com/verne/2015/10/21/articulo/1445432697_658633.html> [consulta: 30 de Julio].

Jhony Auquilla (2013), ¿Qué es el dibujo? [en línea] <<https://es.slideshare.net/JhonnyAuquilla/que-es-el-dibujo>> [consulta: 29 de Julio de 2020].

Jonathan Stonesifer (2018), Using Image Compare with Computer Vision for Testing [en línea] <<https://www.fisglobal.com/-/media/fisglobal/files/pdf/white-paper/ai-image-compare-white-paper.pdf>> [consulta: 28 de Julio de 2020].

Kulas Christian, Sanador Chirstian, Klinker Gudrun (2018), Towards a Development Methodology for Augmented Reality User Interfaces [en línea] <<https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.5.9288&rep=rep1&type=pdf>> [consulta: 3 de Mayo de 2021].

Rosebrock Adrian (2021), How-To: Python Compare Two Images [en línea] <<https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/>> [consulta 1 de Julio de 2021].

Ruiz Torres David (2014), La Realidad Aumentada y su dimensión en el arte: La obra aumentada. [en línea] <<https://revistas.um.es/reapi/article/view/146261>> [consulta 29 de Julio de 2020].

López David (2021), ¿Qué es y para qué sirve el boceto en diseño e ilustración? [en línea] <<https://www.pinayu.com/blog/el-boceto-que-es-para-que-nos-sirve>> [consulta: 3 de junio de 2021]

Marker Graciela (2020), Programas orientados al dibujo [en línea] <<https://www.tecnologia-informatica.com/programas-para-dibujar/>> [consulta: 15 de Septiembre de 2020].

Ramonfaur Garza Bernardo (2018), Dibujo artístico – Nociones básicas [en línea] <https://upload.wikimedia.org/wikipedia/commons/c/cc/Dibujo_art%C3%ADstico_-_Nociones_b%C3%A1sicas.pdf> [consulta: 3 de Junio de 2021]

Real Academia Española (2020), Diccionario de la lengua española [en línea] <<https://dle.rae.es/lenguaje?m=form>> [consulta: 27 de Julio de 2020].

Sanz Fernández Jorge (2019), Como aprender a dibujar utilizando trazos virtuales [en línea] <https://www.lasexta.com/tecnologia-tecnoplora/apps/como-aprender-a-dibujar-utilizando-la-realidad-aumentada_201910015d970afa0cf2b6281a5a7f31.html>[consulta: 10 de Septiembre de 2020].

Soto Gonzalo (2018), Desarrollo de una API para datos abiertos [en línea] <<https://riull.ull.es/xmlui/bitstream/handle/915/7106/Desarrollo%20de%20una%20API%20para%20datos%20abiertos.pdf?sequence=1>> [consulta: 20 de Septiembre de 2020]

Universidad de Granada (2021), Contrastes de hipótesis [en línea] <<http://wpd.ugr.es/~bioestad/guia-de-r/practica-6/>> [consulta: 20 de Enero de 2021]

Westreicher Guillermo (2021), Método Científico. Economipedia. [en línea] <<https://economipedia.com/definiciones/metodo-cientifico.html>> [consulta: 20 de Junio de 2021]

Wolfgang Ertel (2017), Introduction to Artificial Intelligence [en línea] <<https://link.springer.com/book/10.1007%2F978-3-319-58487-4>> [consulta: 20 de Septiembre de 2020].

X. Basogain, M. Olabe, K. Espinosa, C. Rouèche y J.C. Olabe (2017), Realidad Aumentada en la Educación: una tecnología emergente [en línea] <<http://files.trendsandissues.webnode.com/200000010-3884839004/educamadrid-2007.pdf>> [consulta: 29 de Julio de 2020].

Zerna Lesly (2020), ¿La inteligencia artificial es el mayor avance del siglo o una amenaza? [en línea] <<https://ed.team/comunidad/la-inteligencia-artificial-es-el-mayor-avance-del-siglo-o-una-amenaza>> [consulta: 24 de Septiembre de 2020].

DOCUMENTACIÓN