



Universidad de las Ciencias Informáticas

Facultad 4

**Herramienta de reconocimiento facial de emociones para
videojuegos.**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Arisdalia Peña Román

Tutor (es): Ms.C Yuniesky Coca Bergoll

Ing. Liliana Rodríguez Martínez

Ing. Laura Elena Santana Rojas

La Habana, noviembre de 2022

Agradecimiento

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis que tiene por título: y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los 16 días del mes de noviembre del año 2022.

Ms.C Yuniesky Coca Bergoll Ing. Liliana Rodríguez Martínez Ing. Laura Elena Santana Rojas



Firma del Tutor



Firma del Tutor



Firma del Tutor

Arisdalia Peña Román



Firma del Autor

Agradecimiento

Agradecimientos

A mis padres por todo el apoyo incondicional para lograr formarme como persona y como profesional. A mis mejores amigos, a los que están y a los que no, pero siempre me han dado su apoyo en los momentos más difíciles cuando pensaba rendirme, a mis compañeros de carrera, compañeros de residencia por sopesar todas mis buenos y malos momentos. A mis profesores de carrera, aún más a los que se quedaron como amigos, a los tutores por ser guías y buenos consejeros. Gracias a todos por dar su granito de arena para contribuir a formarme en la carrera de Ingeniería en Ciencias Informáticas.

Dedicatoria

Dedicatoria

Este trabajo de diploma está dedicado a las personas que siempre me enseñaron que la perseverancia es la clave a toda meta, que no importa cuán difícil sea, siempre y estén las ganas de demostrar que se puede. Dedicado a mis padres que no perdieron las fuerzas y me hicieron entender que solo es un paso más, que tampoco debía perderlas yo.

Resumen

RESUMEN

Actualmente el desarrollo de videojuegos ha presentado un auge en el mercado mundial donde la nueva generación ha demandado un desarrollo eficaz y más llamativo de los nuevos productos. El presente proyecto de investigación propone el desarrollo de una herramienta de reconocimiento facial destinado a la identificación y reconocimiento de emociones en tiempo real como respuesta a videojuegos. El proyecto será llevado a cabo en el Centro de Tecnologías Interactivas de la Universidad de Ciencias Informáticas; con el objetivo principal de incrementar los índices de mejoras colectiva en los videojuegos según la respuesta del usuario, brindando una herramienta que pueda ser utilizada en diversos videojuegos. Utilizando técnicas de detección de rostros, se localizan las principales regiones de la cara y se extraen los puntos de interés de cada una. A partir de estos puntos se detectan los movimientos musculares de la cara (unidades de acción) y en función de ellos un clasificador reconoce la emoción.

Palabras Clave: Aprendizaje, Emociones, Facial, Reconocimiento, Sistema.

Resumen

ABSTRACT

Currently the development of video games has presented a boom in the world market where the new generation has demanded an effective and more striking development of new products. This research project proposes the development of a prototype of a facial recognition tool for the identification and recognition of emotions in real time in response to video games. The project will be carried out at the Center for Interactive Technologies of the University of Informatics Sciences; with the main objective of increasing the collective improvement rates in video games according to the user's response, providing a tool that can be used in various video games. Using face detection techniques, the main regions of the face are located and points of interest are extracted from each one. From these points, the muscular movements of the face (action units) are detected and based on them a classifier recognizes the emotion.

Keywords: *Learning, Emotions, Facial, Recognition, System.*

Índice

Índice

Introducción	1
CAPÍTULO 1. Fundamentación teórica sobre el reconocimiento facial aplicado a videojuegos.	2
1.1 Conceptos asociados al Reconocimiento Facial	2
1.1.2 Emociones	2
1.1.3 Computación Afectiva.....	3
1.1.4 Reconocimiento facial de emociones.....	3
1.2 Tipos de sistemas de reconocimiento facial	4
1.3 Métodos para el reconocimiento facial de emociones	5
1.4 Técnicas de reconocimiento facial	6
1.4.1 Holísticas.....	6
1.4.2 Geométricas.....	7
1.4.3 Basadas en videos.	8
1.4.4 Selección de la técnica a emplear en la solución.....	8
1.5 Análisis de homólogos	8
1.5.1 "Patrulla de medianoche" (<i>Midnight patrol</i>)	8
1.5.2 Azure Cognitive Services.....	9
1.5.3 Amazon Rekognition	10
1.5.4 Google Visio	10
1.5.5 IBM WATSON	11
1.5.6 Aportes a la solución	11
1.6 Metodología, herramientas y tecnologías.....	12
1.6.1 Metodología de Investigación	12
1.6.2 Lenguajes.....	14
1.6.2.1 Lenguaje de modelado.....	14
1.6.1.2 Lenguaje de programación Python.....	15
1.6.2 Marco de trabajo	15
1.6.3 Herramienta de modelado (<i>Visual Paradigm</i>).....	16
1.6.4 Entorno de Desarrollo Integrado (IDE) PyCharm.....	16
1.6.5 Herramientas de gestión de base de datos	17
1.6.5.1 Bases de datos de emociones	17
1.6.5.2 MySQL integrado en XAMPP versión 3.3.0.....	17
1.6.6 Bibliotecas	17
1.7 Conclusiones del capítulo.....	19

Índice

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN	2
2.1 Particularidades de entrenamiento de una inteligencia artificial para el desarrollo de reconocimiento facial de emociones	2
2.1.1 Esquema general de la aplicación.....	2
2.1.2 Detección y seguimiento de la cara.....	3
2.1.3 Extracción de características	5
2.1.4 Clasificadores	6
2.2 Levantamiento de requisitos.....	8
2.2.1 Requisitos funcionales.....	8
2.2.2 Requisitos no funcionales.....	9
2.3 Historia de usuario	10
2.4 Desarrollo del plan de iteraciones.....	11
2.5 Plan de duración de las iteraciones	12
2.6 Plan de entregas	13
2.7 Arquitectura de la solución	13
2.7.1. Arquitectura de diseño a utilizar	14
2.7.2 Diagrama de componente.....	16
2.8 Tarjetas de Contenido, Responsabilidad y Colaboración.....	16
2.9 Conclusiones parciales.....	17
CAPÍTULO 3: IMPLEMENTACIÓN DE LA HERRAMIENTA DE RECONOCIMIENTO FACIAL Y PRUEBAS REALIZADAS	48
3.1. Descripción y funcionamiento de la herramienta de reconocimiento facial de emociones.....	48
3.1.2 Procesado de la imagen	48
3.1.3 Detección de la cara	49
3.1.4 Pre procesado	50
3.1.5 Localización de regiones y búsqueda de los puntos de interés.	50
3.2 Tareas de ingeniería.....	51
3.2.1 Iteración I.....	51
3.2.2 Iteración II.....	52
3.2.3 Iteración III.....	53
3.2.4 Iteración IV.....	53
3.3 Implementación.....	54
3.4 Diagramas de funcionamiento	57
3.4.1 Diagrama de clases	57

Índice

3.4.2 Diagrama de flujo.....	58
3.5 Resultados obtenidos.....	59
3.6 Pruebas.....	60
3.6.1 Pruebas de aceptación	60
3.7 Conclusiones parciales.....	68
Conclusiones.....	70
Recomendaciones.....	70
Referencias Bibliográficas.....	2
Anexos	2

Índice

Índice de figuras

Figura 2 Fotografía de anuncio sobre la IA desarrollada por Tencent	9
Figura 3 Fotografía de publicidad sobre reconocimiento facial de Microsoft Azure Face Services.....	10
Figura 4 Logo de Amazon Rekognition	¡Error! Marcador no definido.
Figura 5 Sitio Amazon Rekognition	10
Figura 6 Sitio de Google Visio.....	11
Figura 7 Funcionamiento de IBM WATSON.....	11
Figura 8 Esquema general de reconocimiento facial de emociones.....	3
Figura 11 Ecuación 1 umbral de comparación de pixeles vecinos.....	4
Figura 12 Funcionamiento LBP.....	4
Figura 13 Ecuación 2 para la obtención de combinaciones LBP	5
Figura 20 Diagrama de componente.....	16
Figura 22 Detección del rostro.	49
Figura 1. Representación de 68 marcas faciales en el rostro humano.	2
Figura 9 Métodos de detección facial. Fuente Zapatero,2016	2
Figura 10 Algoritmo Viola-Jones.	2
Figura 19 Ciclo de vida de la arquitectura cliente servidor.....	4
Figura 21 Etapas proceso de la imagen.....	5
Figura 23 Etapas fundamentales para el pre procesamiento de una imagen.	5
Figura 24 Resultados obtenidos luego del análisis del rostro.....	5

Índice

Índice de tablas

Tabla 1 Tabla comparativa de soluciones homólogas	12
Tabla 2 Comparación metodologías de software.	13
Tabla 3 Ejemplo de unidades de acción y posible combinación	5
Tabla 4 Historia de usuario # 1	10
Tabla 5 Historia de usuario # 2	11
Tabla 6 Historia de usuario # 3	6
Tabla 7 Historia de usuario # 4	6
Tabla 8 Historia de usuario # 5	7
Tabla 9 Historia de usuario # 6	7
Tabla 10 Historia de usuario # 7	8
Tabla 21 Tarea de ingeniería # 3	11

Introducción

Introducción

Uno de los objetivos en la interacción hombre-máquina consiste en conseguir una comunicación natural y sin esfuerzo. Las nuevas tecnologías permiten que, más allá de la interacción existente a través del teclado y del ratón, surjan nuevas modalidades de comunicación entre un ordenador y el usuario que lo maneja. Dos de los canales más informativos para la percepción de emociones por parte de una máquina son las expresiones faciales obtenidas a partir de un video y las expresiones léxico-fonéticas obtenidas de un discurso. Para que el computador consiga establecer una interacción adecuada ha de ser capaz de tener alguna percepción del estado emocional del ser humano con el que interacciona. La expresión facial de las emociones tiene una función social importante que facilita la interacción entre las personas. Este proceso tiene una base neurológica, que no se aísla del contexto ni de la experiencia acumulada por la interacción entre las personas en dicho contexto. Sin embargo, hasta la fecha, no se conocen con claridad los efectos de la experiencia sobre la percepción de las emociones.

Generalmente, esta identificación de emociones es usada para mejoras de calidad de una aplicación basándose en estadísticas, dígame un videojuego. La industria de los videojuegos es una de las de mayor éxito y con más usuarios en el mundo, y esto se debe, en gran parte, a la alta calidad de las experiencias que ofrecen. Los videojuegos han evolucionado mucho en la última década, pero el uso y desarrollo del reconocimiento facial y de voz está cambiando por completo la experiencia de jugar videojuegos, haciendo que estos sean mucho más realistas, inmersivos e interactivos. Además, de tener excelentes gráficos y una trama interesante, un factor importante para el éxito de los juegos ha sido el uso de tecnologías como el reconocimiento facial para mejorar la forma en que las personas viven el juego. El reconocimiento facial de emociones es una tecnología capaz de identificar o verificar una emoción de un usuario a través de una imagen, vídeo o cualquier elemento audiovisual de su rostro. Es una forma de identificación biométrica que se sirve de medidas corporales, en este caso la cara y cabeza, para identificar la emoción de una persona cuando interactúa con una aplicación determinada. La tecnología recoge un conjunto de datos biométricos únicos de cada persona asociados a su rostro y expresión facial para identificar mediante el uso de la inteligencia artificial el tipo de emoción (Diego, «*et al.*», 2006).

Introducción

Actualmente este método de reconocimiento se está empleando para imitar las características faciales de las personas en los juegos y así poder tener un avatar realmente personalizado que sea lo más fiel a las expresiones reales. También, los desarrolladores esperan que, con tecnología y junto a unas cámaras 3D especiales, se realicen juegos que recreen de manera muy detallada las expresiones de un jugador. Para lograr esto, se deben escanear diferentes puntos del rostro de la persona, y así poder mostrar el aspecto y emociones. Además, este tipo de tecnología ya se usa en algunas plataformas de juegos como método de seguridad, tales como: algunos sitios de videojuegos para PC y los actuales casinos *online*, los cuales también tienen otras innovaciones que mejoran la experiencia de los usuarios como juegos en vivo con un crupier real, haciendo que el juego sea mucho más interactivo y real.

En los últimos años se ha desarrollado un creciente interés en mejorar todos los aspectos de la interacción entre humanos y computadoras. Muchos estudios para distinguir emociones faciales han favorecido la creación de entornos para esta interacción. Especialmente en el área del reconocimiento de emociones humanas al observar expresiones faciales. Gracias al uso de las tecnologías de inteligencia artificial (IA) y *machine learning*, los sistemas de reconocimiento facial pueden funcionar con los más altos estándares de seguridad y fiabilidad. Del mismo modo, gracias a la integración de estos algoritmos y técnicas informáticas, el proceso puede llevarse a cabo en tiempo real. En Cuba se ha estado investigando sobre cómo desarrollar tecnología y dispositivos capaces de reconocer, interpretar y simular las emociones humanas, esto, con la finalidad de diseñar tecnología que simule y realice las funciones propias de una persona, brindando al usuario un ambiente personalizado y adaptado a sus necesidades cognitivas y emocionales. Las diferentes universidades del país, dígase la Universidad de la Habana, la Universidad Tecnológica de La Habana José Antonio Echeverría, la Universidad Central "Marta Abreu" de Las Villas y La Universidad de Ciencias Informáticas, que llevan méritos tanto nacionales como internacionales son algunas de las instituciones que dan a sus estudiantes y al posgrado, tareas que contribuyen al desarrollo de la inteligencia artificial en diferentes temáticas enmarcado estas acciones en el proyecto nuevas tecnologías, industrialización avanzada y reinserción internacional; que da respuesta a uno de los macroprogramas contenidos en el Plan Nacional de Desarrollo Económico y Social hasta 2030.

En Cuba, como parte de la iniciativa de la Universidad de Ciencias Informáticas (UCI) para incursionar en este mercado, el Centro de Tecnologías Interactivas tiene, entre sus proyectos desplegados, disímiles videojuegos. Este centro presenta problemas en

Introducción

dicha área, ya que, a la hora de realizar las pruebas requeridas, estas son ejecutadas en laboratorios de calidad a nivel UCI. Dichas pruebas presentan estándares propios para los softwares, pero los videojuegos poseen particularidades únicas que traen como consecuencia la realización de pruebas no acordes a sus estándares, que los probadores no sepan la forma correcta de llevar a cabo las pruebas y por lo tanto se convierte este proceso de prueba es un proceso mal realizado. El centro actualmente aboga por realizar desde su seno las pruebas de los videojuegos desarrollados teniendo en cuenta que los probadores existentes en el centro tienen experiencia en el área de videojuegos y conocen las métricas a seguir para conocer la calidad de uno. Otro elemento de importancia es que se suele pasar por alto determinar el nivel de satisfacción de los jugadores con los videojuegos y que este proceso de prueba se realiza de una forma muy trabajosa, monótona y lenta.

Partiendo de la situación problemática anterior, se identifica como **problema de investigación**: ¿Cómo mejorar el proceso de pruebas orientado a videojuegos, desarrollados en el Centro de Tecnologías Interactivas?

Para llevar a cabo la investigación y arribar a una solución, se define como **objeto de estudio**: Reconocimiento facial. El **objetivo general** que tendrá la presente investigación será: Desarrollar una herramienta de reconocimiento facial de emociones que determine la respuesta y satisfacción de los usuarios, contribuyendo a mejorar el proceso de pruebas a un videojuego.

Dentro del objeto de estudio de la investigación se precisa como **campo de acción**: el reconocimiento facial de emociones aplicado a videojuegos.

La investigación se orientó a través de las siguientes tareas:

- Análisis de los principales conceptos, contribuciones y tendencias sobre herramientas de reconocimiento facial para determinar la respuesta de usuario ante un videojuego.
- Identificación de las tecnologías, herramientas y técnicas disponibles para el reconocimiento facial de emociones.
- Definición de los aspectos que distinguirán la propuesta de solución.
- Implementación de la solución informática.
- Evaluación de la propuesta de solución aplicando pruebas de aceptación.

A continuación, se mencionan los métodos teóricos y empíricos que fueron empleados durante el desarrollo de la investigación:

Introducción

Métodos teóricos

- **Histórico-Lógico:** Permitió realizar un estudio de los principales conceptos asociados al reconocimiento facial.
- **Hipotético-deductivo:** Se utilizó para guiar la investigación desde el planteamiento del problema hasta la verificación de la solución a partir de las validaciones, orientando la secuencia lógica de las tareas que se realizaron.
- **Análisis y Síntesis:** Se utiliza para identificar y analizar las diversas funcionalidades de las herramientas de reconocimiento facial en videojuegos.
- **Modelación:** Permitió la creación del modelo conceptual para entender el contexto en el que se enmarca la investigación.

Estructura del documento

Capítulo 1. Fundamentación teórica sobre el reconocimiento facial aplicado a videojuegos. En este capítulo se definieron algunos conceptos, los tipos de reconocimientos facial, los métodos para el reconocimiento facial de emociones y técnicas de reconocimiento facial importantes para la posterior comprensión de la investigación. Se realizó una breve reseña y valoración sobre las soluciones existentes de los sistemas de reconocimiento facial y su integración a los videojuegos, y se explican las herramientas, metodologías y lenguajes que serán utilizadas en la construcción de la solución.

Capítulo 2. Planificación y diseño de la propuesta de solución. En este capítulo se definió la solución que se propone y se realizó la selección de los requerimientos del sistema que se pretende implementar. Se hizo el análisis de la solución propuesta, se modelan y describen los recursos ingenieriles necesario acorde a la metodología XP que representan las funcionalidades del sistema, aplicando los patrones de arquitectura y diseño seleccionados.

Capítulo 3. Implementación de la herramienta de reconocimiento facial y pruebas realizadas. En este capítulo se describió la implementación y posterior validación realizada al producto obtenido como solución. Se deja plasmado cada paso que se realizó para construir la herramienta y las pruebas realizadas a la misma.

Capítulo 1

CAPÍTULO 1. Fundamentación teórica sobre el reconocimiento facial aplicado a videojuegos.

Durante este capítulo se abordarán los conceptos fundamentales que serán tratados en el desarrollo de la investigación y en la implementación de la solución. Se analizará la fundamentación teórica del trabajo, incluyéndose en el mismo, el estudio de otras soluciones informáticas existentes en la actualidad y los avances destacados respecto a la inteligencia artificial, las técnicas de *machine learning* relacionadas con el entrenamiento de la inteligencia artificial y las estrategias de reconocimiento facial de emociones que pudieran adaptarse al objeto de estudio. Se describirá la metodología, las herramientas y modelos de desarrollo de software a desarrollar.

1.1 Conceptos asociados al Reconocimiento Facial.

1.1.2 Emociones

Las diferentes concepciones de las emociones se destacan por los acentos puestos en las distintas cualidades de las mismas. No siempre se excluyen explícitamente entre sí, pero tampoco existe una definición unívoca aceptada.

La Real Academia Española define la emoción como: “alteraciones del ánimo intensas y pasajeras, agradables o penosas que van acompañadas de cierta conmoción somática” (Arboleda Rodríguez, Gallar Pérez and Barrios Queipo, 2017). La definición psicológica plantea: “las emociones son reacciones psicológicas que representan modos de adaptación a ciertos estímulos del individuo” (Peinador Yubero, 2016).

El psicólogo Paul Ekman enuncia un concepto moderno y abarcador acerca de la emoción definiéndola como: “Un proceso de tipo particular de valoración automática influida por el pasado evolutivo y personal, en el que se siente que está ocurriendo algo importante para el bienestar, produciendo un conjunto de cambios físicos y comportamentales para hacerse cargo de una situación” (Ekman and Oster, 1979).

De acuerdo con las definiciones anteriores se define la emoción como un fenómeno complejo multidimensional que afecta a los sistemas fisiológicos, conductuales y cognitivos, de origen innato o influenciado por la experiencia y que contribuye a dirigir y organizar la conducta. El psicólogo y antropólogo estadounidense Dr. Paul Ekman publicó con su colega Wallace Friesen en 1978 el *Facial Action Coding System* (Ekman and Friesen, 1978); en este postulan la existencia de seis emociones básicas en todos los seres humanos independiente de las raíces culturales cuya expresión se proyecta en el área facial. Las emociones básicas según ellos serían **alegría, tristeza,**

Capítulo 1

enojo, sorpresa, miedo y asco, expuesto por Arboleda Rodríguez, Gallar Pérez y Barrios Queipo (2017). Estas seis emociones universales definidas por Paul Ekman proporcionan la base de todos los estudios que se desarrollan hoy en día en el reconocimiento automático de expresiones faciales y en toda el área de trabajos realizados en la Computación Afectiva

1.1.3 Computación Afectiva

La Computación Afectiva (*Affective Computing*) hace referencia al diseño de sistemas y dispositivos que pueden reconocer, interpretar, procesar y generar emociones humanas para mejorar la interacción entre el usuario y la computadora (Baldasari, 2016). La Dra. Rosalind W. Picard profesora e investigadora del Instituto Tecnológico de Massachusetts (MIT) fue la primera en acuñar el término. Lo definió como una disciplina dentro del campo de la Inteligencia Artificial que intenta desarrollar métodos de computación focalizados en reconocer las emociones humanas y generar emociones sintéticas (Arboleda Rodríguez, Gallar Pérez and Barrios Queipo, 2017). Esta área se ocupa del reconocimiento de emociones (y de expresiones emotivas) humanas por parte de una computadora y la simulación (o generación) de estados y expresiones emocionales con computadoras. Esta área permite la interpretación de la emoción, el estado de ánimo, la actitud y es capaz de suplir las preferencias y necesidades del usuario. Es una parte esencial dentro de la interacción Humano Computadora (HCI) (Sokolova and Fernández Caballero, 2015). Los seres humanos son capaces de expresar el afecto a través de una serie de canales como la expresión facial, los movimientos corporales, comportamiento de la voz y otras señales fisiológicas, como la frecuencia cardíaca y el sudor, (Tao and Tan, 2014).

1.1.4 Reconocimiento facial de emociones

Es una manera de identificar o confirmar las emociones de una persona mediante su rostro. Los sistemas de reconocimiento facial se pueden utilizar para identificar las emociones de las personas en fotos, videos o en tiempo real. De lo antes expuesto aparece lo que se conoce como sistema de reconocimiento facial de emociones; aplicación dirigida por ordenador que identifica automáticamente a una persona y las emociones de la misma en una imagen digital; esto es posible mediante un análisis de las características faciales del sujeto extraídas de la imagen o de un fotograma clave de una fuente de vídeo, y comparándolas con una base de datos. La extracción de características se hace mediante las unidades de acción que no es más que todos los posibles movimientos faciales detectados visualmente, las funciones de extracción de

Capítulo 1

características permiten codificar las unidades de acción en varias categorías de movimientos y posiciones de ojos y cabeza (Castrillón, Álvarez and López,2008) .

1.2 Tipos de sistemas de reconocimiento facial

El reconocimiento facial de emociones es un método de identificación biométrica que utiliza medidas corporales, en este caso cara y cabeza, para verificar la identidad de una persona a través de su patrón biométrico facial; donde se recopila un conjunto de datos biométricos únicos de cada persona asociados con su rostro y expresión facial para identificar, verificar y / o autenticar a una persona (Dictamen 3 ,2012).

Existen principalmente dos tipos de sistemas de reconocimiento facial, aunque también existen sistemas mixtos, que engloban las siguientes características (Reconocimiento facial, EcuRed, s.f.):

- Aquellos basados en los rasgos faciales del individuo tales como: ojos, nariz, boca, otras. Utilizan además mediciones de distancias entre los distintos rasgos y mediciones de los ángulos de la cara. Son los llamados sistemas geométricos.
- Los sistemas fotométricos, que aportan información de forma global, del rostro completo.

Los sistemas tecnológicos a veces pueden variar en lo que respecta al reconocimiento facial, pero el funcionamiento general es el siguiente:

- **Detección de rostro:** La cámara detectará y reconocerá un rostro, ya sea solo o entre una multitud. La cara se detecta mejor cuando la persona mira directamente a la cámara. Los avances tecnológicos han permitido que ligeras variaciones de esto también funcionen.
- **Análisis facial:** Se captura y analiza una foto del rostro. La mayoría del reconocimiento facial se basa en imágenes 2D en lugar de 3D porque puede hacer coincidir más convenientemente una foto 2D con fotos públicas o con las de una base de datos. Hitos distinguibles o puntos nodales componen cada cara. Cada rostro humano tiene 80 puntos nodales. El software de reconocimiento facial analizará los puntos nodales como la distancia entre los ojos o la forma de los pómulos.
- **Convertir una imagen en datos:** El análisis del rostro se convierte luego en una fórmula matemática. Estos rasgos faciales se convierten en números en un

Capítulo 1

código. Este código numérico se llama huella. Similar a la estructura única de una huella digital, cada persona tiene su propia huella facial.

- **Comparar imágenes:** Su código se compara con una base de datos de otras impresiones *faceprints*. Esta base de datos tiene fotografías con identificación que se pueden comparar.

1.3 Métodos para el reconocimiento facial de emociones

El reconocimiento facial cuenta con métodos que utilizan imágenes de intensidad. Estos métodos están basados en enfoques los cuales agrupan técnicas para el desarrollo de aplicaciones que utilicen el desarrollo de una inteligencia artificial. Los métodos utilizados en aplicaciones de este tipo, pueden ser más completos que otros en cuanto a la calidad de análisis del rostro en tiempo real, ya que para que estas aplicaciones se desarrollen con altos grados de eficiencia hay que tener en cuenta factores como la iluminación, fondo de la imagen o video y movimientos de la cara (López,2014).

- **Métodos heurísticos:** los enfoques heurísticos se dividen comúnmente en tres tipos: el enfoque no basado en contenido, el enfoque basado en contenido y el enfoque basado en similitud visual, siendo este último el más utilizando (Hernández Domínguez & Baluja García, 2021).
- **Métodos de aprendizaje automático (Machine Learning):** en este método se encuentran agrupadas técnicas de la Inteligencia Artificial para reconocer un rostro en dependencia de la emoción que se detecte. Estas son los Árboles de Decisión, los Métodos de Conjunto, los Modelos Probabilísticos, la Máquina de Soporte Vectorial, la Lógica Difusa las Redes Neuronales y los algoritmos de Aprendizaje profundo. De cada uno de los métodos de Aprendizaje Automático se derivan diversos enfoques que son aplicados en los sistemas reconocimiento facial, por lo que uno de los factores analizados siempre es el nivel de efectividad que estos tienen (Aprendizaje automático, EcuRed, s.f.).
- **Sistemas expertos:** Están compuestos por dos partes principales, el ambiente de desarrollo y el ambiente de consulta. El ambiente de desarrollo es utilizado por el constructor para crear los componentes e introducir conocimiento en la base de conocimiento. El ambiente de consulta es utilizado por los no-expertos para obtener conocimiento experto y consejos (Turban, 1995).
- **Máquina de Estados Finitas:** Se definen como un conjunto de estados que sirven de intermediarios en esta relación de entradas y salidas, haciendo que el

Capítulo 1

historial de señales de entrada determine, para cada instante, un estado para la máquina de forma tal que la salida depende únicamente del estado y las entradas actuales (Casanovas, 2014).

- **Árbol de comportamiento:** Es un modelo matemático de ejecución de planes que se utiliza en informática, robótica, sistemas de control y videojuegos. Describen cambios entre un conjunto finito de tareas de forma modular. Su fortaleza proviene de su capacidad para crear tareas muy complejas compuestas por tareas simples, sin preocuparse de cómo se implementan las tareas simples. Los árboles de comportamiento presentan algunas similitudes con las máquinas de estado jerárquicas, con la diferencia clave de que el componente principal de un comportamiento es una tarea en lugar de un estado. Su facilidad de comprensión humana hace que los árboles de comportamiento sean menos propensos a errores y muy populares en la comunidad de desarrolladores de juegos. Se ha demostrado que los árboles de comportamiento generalizan varias otras arquitecturas de control. Matemáticamente, son gráficos acíclicos dirigidos (Árbol de comportamiento (inteligencia artificial, robótica y control),s.f).

1.4 Técnicas de reconocimiento facial

Las técnicas utilizadas dentro del dominio de la Inteligencia Artificial son en realidad sólo formas avanzadas de modelos estadísticos y matemáticos. Todos estos modelos conjuntos inteligentemente proporcionan herramientas para calcular tareas que antes se pensaba que estaban reservadas para los seres humanos.

1.4.1 Holísticas

En este enfoque, la región de la cara completa se tiene en cuenta como datos de entrada en el sistema de captura de rostros. Uno de los mejores ejemplos de métodos holísticos son las caras propias, PCA, análisis discriminante lineal y análisis de componentes independientes.

Los pasos en este método son los siguientes:

- Insertar un conjunto de imágenes en una base de datos, estas imágenes se nombran como el conjunto de entrenamiento porque se utilizarán cuando se comparan imágenes y se crean las caras propias.

Capítulo 1

- Las caras propias se crean extrayendo rasgos característicos. Las imágenes de entrada se normalizan para alinear los ojos y la boca. Luego se les cambia el tamaño para que tengan la misma dimensión. Las caras propias ahora se pueden extraer de los datos de la imagen utilizando una herramienta matemática llamada PCA.
- Ahora cada imagen se representará como un vector de pesos. El sistema ahora está listo para aceptar consultas. El peso de la imagen desconocida entrante se encuentra y luego se compara con los pesos de las imágenes ya presentes en el sistema.
- Si el peso de la imagen de entrada está por encima de un umbral determinado, se considera no identificado. La identificación de la imagen de entrada se realiza encontrando la imagen en la base de datos cuyos pesos son los más cercanos a los pesos de la imagen de entrada.
- La imagen en la base de datos con el peso más cercano se devolverá como un acierto al usuario.

1.4.2 Geométricas.

En el reconocimiento facial 2D, las imágenes a menudo se representan por su estructura geométrica o codificando sus valores de intensidad. Se obtiene una representación geométrica transformando la imagen en primitivas geométricas como puntos y curvas. Esto se hace, por ejemplo, localizando rasgos distintivos como ojos, boca, nariz y mentón, y midiendo su posición relativa, ancho y otros parámetros.

La representación basada en la apariencia se basa en registrar varias estadísticas de los valores de los píxeles dentro de la imagen de la cara. Los ejemplos incluyen: registrar las intensidades de la imagen como matrices 2D llamadas plantillas y calcular histogramas de las salidas de los detectores de bordes (Cadena,2021).

El reconocimiento facial 3D es un método de reconocimiento facial que aprovecha la información geométrica 3D del rostro humano. Emplea datos de sensores 3D que capturan información sobre la forma de una cara. El reconocimiento se basa en la coincidencia de metadatos extraídos de las formas 3D de las caras. En un escenario de identificación, la coincidencia es de uno a muchos, en el sentido de que una sonda se compara con todos los datos de la galería para encontrar la mejor coincidencia por encima de algún umbral.

Capítulo 1

1.4.3 Basadas en videos

El reconocimiento facial basado en video es la técnica de establecer la identidad de una o varias personas presentes en un video, en función de sus características faciales. Dado el video facial de entrada, un enfoque típico de reconocimiento facial en video combina las características temporales del movimiento facial con cambios de apariencia para el reconocimiento. Esto a menudo implica la caracterización temporal de rostros para el reconocimiento, construyendo un modelo 3D o una imagen de super resolución de la cara, o simplemente aprendiendo las variaciones de apariencia de los múltiples cuadros de video (Pérez,2013).

1.4.4 Selección de la técnica a emplear en la solución

Después del análisis de las técnicas existentes para el reconocimiento facial de emociones se concluyó que la mejor técnica a utilizar será la técnica geométrica en 2D. Esta técnica para el análisis de características faciales es más efectiva en cuanto hacer coincidir los puntos faciales previamente definidos para la extracción de características en el rostro. De esta forma que será más sencillo poder determinar la emoción que se va a detectar haciendo un conjunto de combinaciones de las unidades de acción del rostro.

1.5 Análisis de homólogos

Dado el grado de dinamismo presente en el campo del reconocimiento facial, es común encontrar varias formas de aproximarse a la solución de un mismo problema. A continuación, se describe una representación de programas que utilizan el reconocimiento facial en videojuegos. Las características tomadas en cuenta para el análisis fueron: rasgos faciales, clasificación de las emociones, análisis de imagen o video por parte del sistema, sistema operativo o dispositivo para montaje de la aplicación, compra de la aplicación o pago por utilizar el sistema y tipo de software.

1.5.1 "Patrulla de medianoche" (*Midnight patrol*)

Tencent Games la desarrolladora de China, famosa por títulos como *PUBG* (además de por aplicaciones como *WeChat*), introdujo en sesenta de sus títulos una herramienta de escaneo facial. Todo para detectar si los niños usan los juegos móviles por la noche y, de paso, bloquearlos. Se encarga de interrumpir la partida durante las horas restringidas (de 10 de la noche a 8 de la mañana) para analizar mediante

Capítulo 1

escaneo facial si la cara corresponde con un adulto autorizado. En caso de que el jugador sea menor, y ya haya superado sus 90 minutos de juego (durante los festivales el tiempo se amplía hasta las tres horas), el título quedará bloqueado hasta el día siguiente.



Figura 1 Fotografía de anuncio sobre la IA desarrollada por Tencent.

A excepción de este programa que usa el reconocimiento facial para un propósito muy diferente al que se quiere llegar con esta investigación, todas las demás aplicaciones que se investigaron posteriormente solo contribuyen al reconocimiento facial de emociones en diversas áreas que no incluyen la rama de los videojuegos.

1.5.2 Azure Cognitive Services.

Microsoft Azure Face Services, dispone de servicios de reconocimiento facial, detección de caras y detección de emociones a través de *Azure Cognitive Services*. No dispone de posibilidad de prueba *online* a la API, pero permite realizar pruebas de manera gratuita y pone a la disposición de todos repositorios de código con ejemplos de uso (Author Microsoft).



Figura 2 Fotografía de publicidad sobre reconocimiento facial de Microsoft Azure Face Services.

1.5.3 Amazon Rekognition

Amazon con su *Amazon Rekognition* ofrece la posibilidad de además de analizar emociones detectar textos, etiquetar imágenes, identificar contenidos, detección de equipos de protección profesional y detectar famosos. No se puede realizar una prueba *online* a la API, pero permite realizar de manera gratuita 1000 usos por mes tanto para reconocer 5.000 imágenes por mes. Funciona igual para vídeo como para imágenes (Author Amazon).

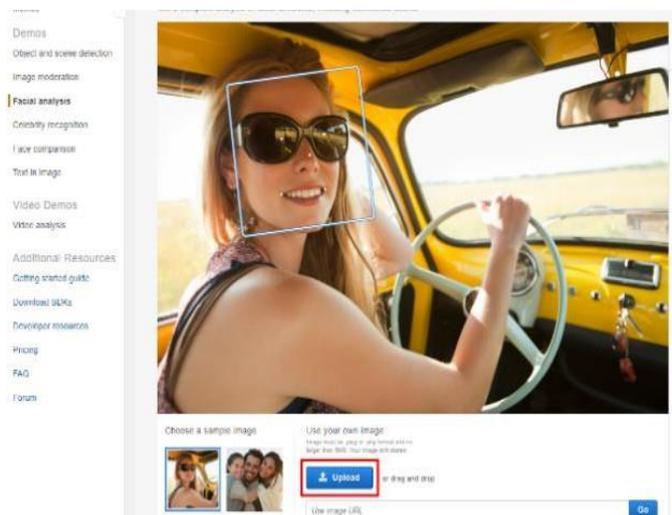


Figura 3 Sitio Amazon Rekognition.

1.5.4 Google Visio

Esta tecnología permite detectar rostros, realizar reconocimiento facial y analizar las emociones. Se puede probar de forma online su API de una manera bastante sencilla.

Capítulo 1

Los precios se cobran por uso con una pequeña capa gratuita, y como en todos los proveedores de la nube aparecen siempre costes ocultos que incrementan la factura final (Author Google)

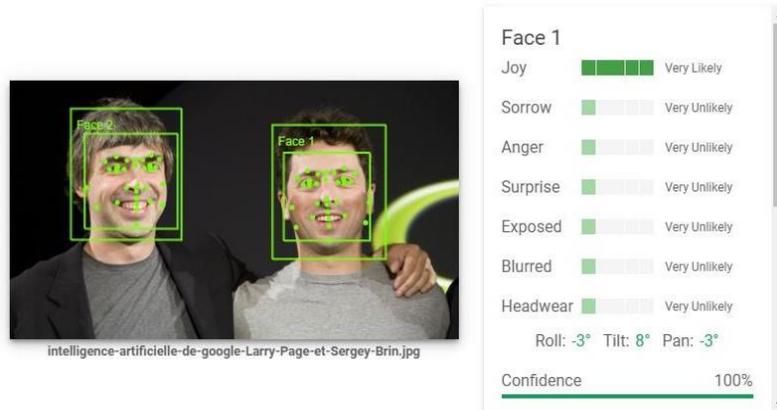


Figura 4 Sitio de Google Visio.

1.5.5 IBM WATSON

IBM sigue la misma filosofía que las anteriores, a través de su famoso Watson ofrece la posibilidad de analizar las emociones humanas en imágenes siguiendo el espíritu de uso habitual de su nube. Dispone de una capa gratuita para pruebas y de facturación por uso (Author OpenExpoEurope).

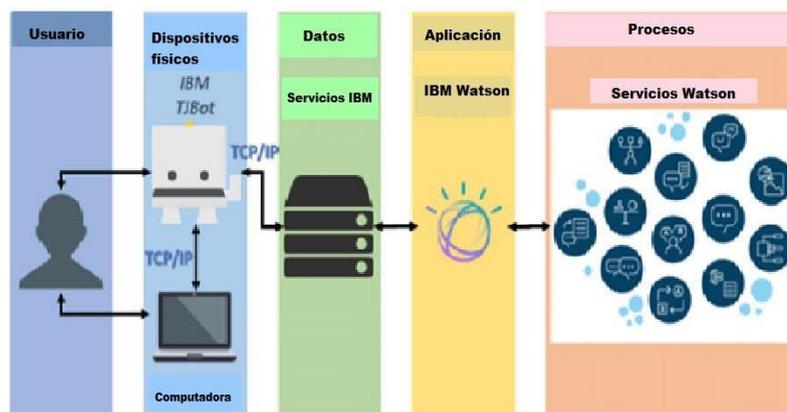


Figura 5 Funcionamiento de IBM WATSON

1.5.6 Aportes a la solución

La Tabla 1 es una comparativa de los diferentes homólogos estudiados en el epígrafe anterior, teniendo en consideración el formato de la aplicación, el tipo de software, si son libres de pago y el sistema operativo donde se pueden desplegar luego.

Capítulo 1

Tabla 1 Tabla comparativa de soluciones homólogas.

Plataforma	Formato de la aplicación	Tipo de Software	Pago por aplicación	Dispositivo o sistema operativo
Patrulla de medianoche (<i>Midnight patrol</i>)	Aplicación android	Privativo	Pago por consumo	Android
Azure Cognitive Services.	Aplicación web	Privativo	Pago por consumo	Windows, Linux
Amazon Rekognition	Aplicación web	Privativo	Pago por consumo	Basada en la nube
Google Visio	Aplicación web	Privativo	Pago por consumo	Windows, Linux
IBM Watson	Aplicación web	Privativo	Pago por consumo	Software DeepQA, Linux

Luego de realizada la comparativa anterior se llegó a la conclusión de que ninguno de los sistemas homólogos da respuesta directa a la situación problemática planteada, pero, aportaron una guía de pasos a seguir para obtener una emoción a partir de técnicas de reconocimiento facial.

1.6 Metodología, herramientas y tecnologías

En el siguiente epígrafe se describe la metodología utilizada para guiar el proceso de desarrollo del software y las diferentes herramientas y tecnologías que se utilizaron para darle solución a la problemática

1.6.1 Metodología de Investigación

Piattini define a la metodología de desarrollo de software como “un conjunto de procedimientos, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”. La metodología indica cómo hay que obtener los diferentes productos parciales y finales. Las metodologías se clasifican en 2 grupos: ágiles y tradicionales. A continuación, se muestran algunas características (Pianttini,1996):

Capítulo 1

Tabla 2 Comparación metodologías de software.

Metodologías ágiles	Metodologías tradicionales
Preparadas para cambios en el proyecto.	Cierta resistencia a los cambios.
No existe un contrato tradicional o al menos es bastante flexible.	Existe un contrato prefijado.
El cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Grupos pequeños (menos de 10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos.	Muchos artefactos
Pocos roles.	Muchos roles.

Luego de realizar un análisis comparativo entre los dos enfoques para metodologías de software, el estudio de sus principales características y ventajas que posibilitan; se determina utilizar una metodología ágil. El equipo de desarrollo es de solo una persona, por lo que se dificulta la aplicación de una metodología tradicional, debido a los puntos expuestos con anterioridad.

Existen diferentes metodologías basadas en la filosofía ágil. Sin embargo, las más utilizadas actualmente son tres: *SCRUM*, *Kanban* y *eXtreme Programming (XP)*.

Dentro de las metodologías ágiles están (LN Creatividad y Tecnología, 2019):

- **Extreme Programming (XP)**, por sus siglas en inglés) Programación Extrema: Es una metodología centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando

Capítulo 1

un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

- **SCRUM:** Describe la forma en que los miembros de equipo deben trabajar para poder obtener un sistema flexible en un entorno que varía de manera constante. Se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.
- **Kanban:** La metodología Kanban se implementa por medio de tableros Kanban. Se trata de un método visual de gestión de proyectos que permite a los equipos visualizar sus flujos de trabajo y la carga de trabajo. En un tablero Kanban, el trabajo se muestra, en un proyecto, en forma de tablero organizado por columnas.

Después de presentadas las características principales de estas tres variantes de la vertiente ágil se decide adoptar XP como metodología para guiar el desarrollo de software, puesto que el flujo de trabajo, además de la simplicidad propuesta por ella, se adaptan a las necesidades del proyecto y las características del equipo de desarrollo. Esta metodología se centra en el testeado para el desarrollo de un producto de software funcional, permitiendo la participación del cliente en todo el proceso como condición fundamental para el resultado exitoso del proceso, promoviendo el trabajo en equipo e impulsando el buen clima laboral. El ciclo de vida establecido para este modelo consta de cuatro fases las cuales son planificación de la entrega, diseño, implementación y pruebas (Pacienza,2015).

1.6.2 Lenguajes

1.6.2.1 Lenguaje de modelado

El lenguaje de modelado a utilizar será UML, el cual es un lenguaje diseñado para visualizar, especificar, construir y documentar software orientados a objeto. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos

Capítulo 1

conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. Se utiliza para especificar o para describir métodos o procesos, para definir un sistema, para detallar los artefactos en el sistema (Hernández, Peña and Rodríguez,2016)

1.6.1.2 Lenguaje de programación Python

Es un lenguaje de código abierto, con una sintaxis muy limpia y sencilla, que favorece un código legible. Se trata de un lenguaje interpretado o de *script*, con tipado dinámico, fuertemente tipado, aunque esto es opcional debido a que se implementa mediante una librería, multiplataforma y orientado a objetos. Además, permite la programación imperativa, funcional y orientada a aspectos. El intérprete de Python está disponible en una multitud de plataformas como UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, entre otros. Es ampliamente utilizado en la industria de los videojuegos, en el desarrollo web y sistemas distribuidos, para el desarrollo de tareas científicas y aplicaciones clientes. Se pueden crear los diagramas de recursos, componentes, despliegue, actividad y el modelo de implementación. La herramienta tiene un buen entorno de trabajo. Facilita la visualización y la manipulación del proyecto de modelado.

1.6.2 Marco de trabajo

Django es un *framework* de desarrollo web de código abierto y multiplataforma. El patrón de Django es MVT – Modelo Vista Plantilla. Debido a que es un *framework* de alto nivel fomenta el desarrollo rápido y el diseño limpio y pragmático. Es muy seguro y ayuda a los desarrolladores a evitar muchos errores comunes de seguridad (Softekk,2018).

Django reduce enormemente el tiempo de desarrollo debido a:

- Un ORM (es un modelo de programación que permite mapear las estructuras de una base de datos relacional (SQL Server, Oracle, MySQL, etc.)) práctico y agradable, con migraciones creadas y aplicadas automáticamente por el framework.
- Andamio para generación automática de panel de administración en base a sus modelos.
- Soporte para cookies, sesiones, middleware y plantillas.

Capítulo 1

- Las funciones de seguridad como prevención XSS (*Cross Site Scripting*), prevención CRSF (*Cross Site Request Forgery*), se aplican automáticamente.

1.6.3 Herramienta de modelado (*Visual Paradigm*)

La herramienta de modelado seleccionada fue *Visual Paradigm* en su versión 15.1 debido a que es una herramienta para el desarrollo de aplicaciones utilizando modelado UML. Esta herramienta es ideal en la construcción de sistemas a gran escala que requieren confiabilidad y estabilidad en el desarrollo orientado a objetos.

- Disponibilidad en múltiples plataformas (Windows, Linux).
- Diseño centrado en casos de uso.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Capacidades de ingeniería directa e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Licencia: gratuita y comercial.
- Varios idiomas.

1.6.4 Entorno de Desarrollo Integrado (IDE) PyCharm

Es un entorno de desarrollo integrado (IDE) utilizado en programación informática, concretamente para el lenguaje de programación *Python*. Está desarrollado por la empresa checa *JetBrains* (antes conocida como *IntelliJ*). Proporciona análisis de código, un depurador gráfico, un probador de unidades integrado, integración con sistemas de control de versiones (VCS), y soporta el desarrollo web con Django.

Razones para utilizar PyCharm:

- Refactorización de Python: incluye renombrar, extraer método, introducir variable, introducir constante, mover hacia arriba, desplazar hacia abajo y otros.
- Soporte para *frameworks* web: Django, web2py y Flask (sólo edición profesional).
- Depurador de Python integrado.
- Pruebas unitarias integradas, con cobertura de código línea por línea.

Capítulo 1

1.6.5 Herramientas de gestión de base de datos

1.6.5.1 Bases de datos de emociones

Las bases de datos de emociones juegan un papel importante en la detección automática de emociones faciales. Son esenciales en la etapa de clasificación. De esta manera la precisión del clasificador para reconocer las emociones depende del conjunto de datos principalmente. Además, los conjuntos de datos definen las emociones que se van a detectar y sistema reconoce esas mismas emociones.

1.6.5.2 MySQL integrado en XAMPP versión 3.3.0

Es el más popular sistema de gestión de base de datos multiplataforma de código abierto. Se caracteriza fundamentalmente por su velocidad, rendimiento y fiabilidad, motivo por el cual es ampliamente utilizado en sitios web y aplicaciones de escritorio de baja y mediana complejidad.

Además, cumple con las siguientes características:

- MySQL es de código abierto.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

1.6.6 Bibliotecas

1.6.6.1 OpenCV (Open Source Computer Vision Library)

Biblioteca de código abierto. Utilizada para el procesamiento de imágenes y el aprendizaje automático. Tiene interfaces C++, Python, Java y MATLAB, y es compatible con los sistemas operativos Windows, Linux, MacOS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en las aplicaciones en tiempo real. La biblioteca está escrita en C / C++ optimizado y hace uso del procesamiento multinúcleo. La biblioteca cuenta con más de 2500 algoritmos optimizados y un gran número de funciones que incluyen un conjunto completo de algoritmos de aprendizaje automático y de visión artificial tanto clásicos como de última generación. Mediante dichos algoritmos se pueden detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, Además, se puede rastrear movimientos de cámara, objetos en movimiento, extraer modelos 3D de objetos,

Capítulo 1

producir nubes de puntos 3D desde cámaras estéreo, unir imágenes para producir una alta resolución. También permite encontrar imágenes similares de una base de datos de imágenes, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos de los ojos, reconocer escenarios y establecer marcadores para superponerlos con realidad aumentada (Softekk,2018).

1.6.6.2 NumPy

Es un paquete de Python que significa “*Numerical Python*”, es la librería principal para la informática científica, proporciona potentes estructuras de datos, implementando matrices simples y multidimensionales. Estas estructuras de datos garantizan cálculos eficientes con matrices. NumPy ocupa menos memoria en comparación a la lista de Python, a su vez es bastante rápido en términos de ejecución.

1.6.6.3 DeepFace

DeepFace es la biblioteca de análisis de atributos faciales y reconocimiento facial más ligera para Python. *DeepFace* es una biblioteca de código abierto que incluye todos los modelos de inteligencia artificial de vanguardia para el reconocimiento facial y maneja automáticamente todos los procedimientos de reconocimiento facial en segundo plano. Los desarrolladores pueden usar, modificar y distribuir la biblioteca en un contexto privado o comercial. Si se ejecuta el reconocimiento facial con *DeepFace*, se tiene acceso a un gran conjunto de funciones que facilitan su uso (Softekk,2018).

1.6.6.4 H5py (Facial Expression Model Weight)

El paquete h5py es una interfaz *Pythonic* para el formato de datos binarios HDF5. Le permite almacenar grandes cantidades de datos numéricos y manipular fácilmente esos datos desde NumPy. Además de la interfaz de alto nivel fácil de usar, h5py se basa en un envoltorio Cython orientado a objetos de la API HDF5 C. Casi todo lo que se puede hacer desde C en HDF5, se puede hacer desde h5py. Los archivos que crea están en un formato binario estándar ampliamente utilizado, que puede intercambiar con otras personas, incluidas aquellas que usan programas como IDL y MATLAB.

1.6.6.5 TensorFlow

Es la tecnología más utilizada (ya sea de manera integral o como base) es *TensorFlowJS*. Es una librería de software orientada al *Machine Learning* y al *Deep Learning*. En sus inicios fue creada por Google, pero hace tiempo que fue liberada y actualmente es desarrollada y mantenida con ayuda de la comunidad (Martinez,2021).

Capítulo 1

TensorFlow funciona en multitud de sistemas operativos y con gran cantidad de lenguajes por lo que su implantación es muy sencilla. Permite generar modelos de aprendizaje con gran facilidad o lo que es mejor, utilizar modelos creados por terceros o por la comunidad.

Hay multitud de proyectos con el código disponible que utilizan *TensorFlow* en *GitHub*. Uno de los más completos y populares es *face-api.js*. Se puede probar todas sus posibilidades online en su web principal. Permite detectar caras, analizar emociones, análisis de vídeo, etc. En este caso el único coste es el servidor donde se quiera correr. El código fuente está abierto, así que se puede modificar lo que se considere necesario.

1.6.6.6 Tkinter

Tkinter es un paquete ("interfaz Tk") es la interfaz estándar de Python para el kit de herramientas GUI Tcl/Tk (*Graphical User Interface*) (*Tool Command Language/Tkinter*). Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, incluido macOS, así como en los sistemas Windows. Tkinter admite una variedad de versiones de Tcl/Tk, construidas con o sin soporte de subprocessos. El lanzamiento binario oficial de Python incluye Tcl/Tk 8.6 subprocessos. No es una sola biblioteca, sino que consta de unos pocos módulos distintos, cada uno con una funcionalidad separada y su propia documentación oficial. Las versiones binarias de Python también incluyen un módulo complementario(Author Python).

Nota: GUI es una interfaz entre la persona y la máquina. El objetivo de esta interfaz gráfica es representar el código del *backend* de un sistema de la forma más clara posible para el usuario para simplificarle las tareas diarias. Para esto, son muy importantes los íconos y las imágenes, ya que solo estos permiten una aplicación universal e independiente del texto(Author Python).

1.7 Conclusiones del capítulo

Durante el presente capítulo, se recopilaron conceptos relacionados con IA de distintos autores contribuyó a consolidar una definición que sirve de base para el resto de la investigación. El análisis de las principales técnicas de IA presentes en el reconocimiento facial y la consulta de soluciones semejantes desarrolladas con anterioridad, resaltaron la efectividad de la técnica geométrica para modelos 2D como base en la búsqueda de soluciones cercanas a las óptimas. Tras realizar una comparación entre metodologías tradicionales y ágiles, se llegó a la conclusión de que

Capítulo 1

estas últimas eran las más adecuadas para guiar el desarrollo e implementación de la propuesta de solución; entre las variantes ágiles, se decidió seleccionar la metodología de software XP. Se seleccionó un conjunto de herramientas para el desarrollo de la aplicación tales como el IDE Pycharm en su versión del 2020 y las bibliotecas de OpenCV , Numpy ,Deepface, TensorFlow ,H5py y Tkinter.

Capítulo 2

CAPÍTULO 2: PLANIFICACIÓN Y DISEÑO DE LA PROPUESTA DE SOLUCIÓN

En este capítulo se exponen las particularidades de entrenamiento de una inteligencia artificial para el desarrollo de reconocimiento facial de emociones y se aborda la propuesta de solución a la problemática planteada. Se describen los requisitos funcionales y no funcionales, para dar cumplimiento a los objetivos planteados. Los requisitos describen las funcionalidades y los atributos de calidad que debe poseer el software. Se realizan las historias de usuario para la descripción detallada de los requisitos funcionales, tarjetas CRC, el plan de iteraciones; mediante el uso de este último, se procederá a la estimación del tiempo requerido para la culminación del desarrollo y demás artefactos exigidos por la Metodología XP y sus fases.

2.1 Entrenamiento de una inteligencia artificial para el desarrollo de reconocimiento facial de emociones

Los sistemas automáticos de reconocimiento facial de emociones tienen como misión detectar la emoción de una persona en función de sus expresiones faciales y en general, suelen estar divididos en varias etapas. Una primera etapa donde se detecta la cara, seguida de otra en la cual se extraen todas las características más significativas y por último una etapa donde un clasificador, que previamente ha sido entrenado, detecta la emoción en función de las particularidades de cada cara.

2.1.1 Esquema general de la aplicación

A continuación, se describe el procedimiento que sigue la aplicación para reconocer una emoción es similar al implementado en cualquier sistema de este tipo:

- Se captura y almacena la imagen neutral una vez que el usuario está preparado.
- En la fase de procesado de la imagen se detecta la cara, se localizan las regiones y se buscan los puntos de interés.
- Posteriormente el usuario ya puede expresar la emoción a detectar para que la aplicación sea capaz de reconocerla. La aplicación analiza todas las imágenes capturadas en tiempo real y realiza los mismos pasos que para la imagen neutral.
- Comparar los puntos de la imagen con emoción y la imagen neutral guardada para detectar las unidades de acción presentes en esa imagen.
- Una vez determinadas esas unidades de acción, se introducen en un clasificador entrenado para que clasifique la emoción.

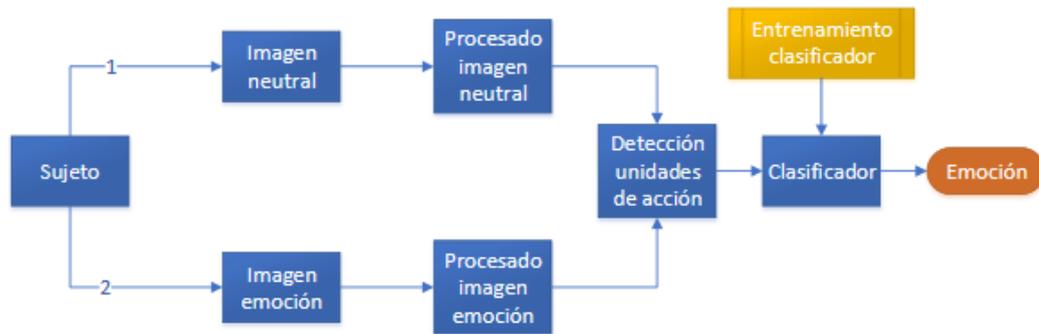


Figura 6 Esquema general de reconocimiento facial de emociones.

2.1.2 Detección y seguimiento de la cara

Esta etapa tiene como fin separar la imagen en dos partes: la cara y el fondo. Los primeros métodos en la detección y seguimiento de la cara se empezaron a desarrollar a partir de principios de la década de 1990. Hjemal and Low reunieron los métodos de detección facial que existían hasta 2001 dividiéndolos en dos grandes grupos: métodos basados en características y métodos basados en imágenes. (E. Hjelmas, B. Kee Low, 2001). Observar figura 9 en la sección de **anexos**.

El método que ha tenido más éxito desde su creación ha sido el desarrollado por Viola y Jones en 2001 y mejorado en 2004. Este método, englobado dentro de los métodos basados en características y en el subgrupo de análisis de características, ha sido frecuentemente utilizado gracias a su rapidez y a los grandes resultados que proporciona en caras de frente. Se basa en el algoritmo de aprendizaje AdaBoost y en la combinación de distintos clasificadores ordenados de menor a mayor complejidad, uno detrás de otro en forma de cascada, que conjuntamente permiten buscar las zonas con mayor cantidad de información útil y descartar las regiones que no son de interés.

Lo primero de todo, los clasificadores, han de ser entrenados con un gran número de muestras del objeto que se quiere detectar. Los clasificadores se basan en características llamadas "*Haar-like features*" como las que se pueden observar en la Figura 10. Los clasificadores van aplicándolas a la imagen para reconocer los bordes, las líneas. Es un proceso que se consigue localizar la cara. Dicho proceso está también explicado y reproducido por Adam Harvey en una entrevista que concedió en 2012. Observar figura 10 en la sección de **anexos**.

Capítulo 2

Otro método es el LBP (*Local Binary Patterns* o Patrones Binarios Locales) introducido en 2002 por Timo Ojala y utilizado en general para detectar todo tipo de objetos basado en texturas, invariante ante cambios de iluminación y capaz de detectar movimiento.

Se trata de un sistema capaz de extraer la estructura local en una imagen a partir del entorno de cada uno de los píxeles que la componen. El proceso consiste en ir comparando el valor de cada pixel (considerado como píxel central) con el de sus vecinos distribuidos en una circunferencia a su alrededor. La comparación se hace en base a la Ecuación (1) siendo x_c el valor central y x el valor a evaluar.

$$s(x) = \begin{cases} 1 & \text{if } (x \geq x_c) \\ 0 & \text{else} \end{cases}$$

Figura 7 Ecuación 1 umbral de comparación de píxeles vecinos.

De tal manera que, si el valor del píxel vecino es mayor que el central entonces el valor será '1', mientras que si es menos será '0' como se puede ver en el ejemplo de Figura 11. De este modo, se asocia un valor binario a cada píxel formando un número binario.

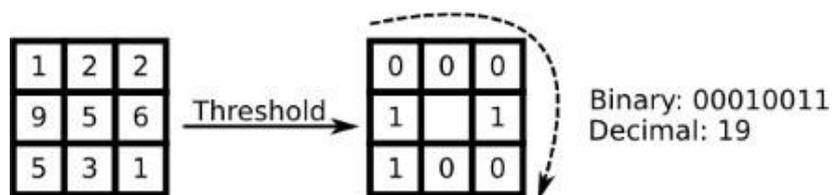


Figura 8 Funcionamiento LBP.

Si hay 8 píxeles vecinos se obtendrá un número de 8 dígitos obteniendo así combinaciones posibles, llamadas LBP, y obtenidas matemáticamente a partir de la expresión de la Ecuación (2), donde i_c hace referencia a la intensidad del punto central y i_p a la intensidad de cada vecino.

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c) \quad (2)$$

Figura 9 Ecuación 2 para la obtención de combinaciones LBP

Al comparar estos dos métodos comentados hasta ahora para detectar caras explicando las ventajas e inconvenientes de cada uno. El algoritmo de Viola Jones es prácticamente inigualable en cuanto a precisión y sobre todo a la ratio de precisión/velocidad. En cambio, aunque el algoritmo de LBP es menos preciso, es más rápido e invariante frente a cambios de iluminación.

Un método muy diferente a estos dos anteriores utilizado sobre todo para el seguimiento de la cara pero que también está teniendo gran repercusión es el método denominado “Candide” basado en modelos parametrizados. La malla se superpone a la cara y se adapta en función del movimiento de las facciones. Este método se compone de una máscara o malla que contiene las facciones de un rostro humano divididas en triángulos como se puede apreciar en la Figura 14 en la sección de **anexos**.

2.1.3 Extracción de características

Se definieron 46 unidades de acción, aunque no todas son igual de representativas puesto que unas aportan más información relevante que otras. En la Tabla 3 se encuentra un ejemplo de algunas unidades de acción. La primera columna se corresponde con el número correspondiente a la unidad de acción, la segunda con su representación y la tercera con una breve descripción de ella. En la cuarta columna se puede observar una imagen donde se combinan esas unidades de acción.

AU	Imagen	Definición	Combinación
AU 1		Interior de las cejas elevado	
AU 2		Exterior de las cejas elevado	
AU 5		Párpado superior elevado	
AU 27		Boca abierta	

Tabla 3 Ejemplo de unidades de acción y posible combinación.

Capítulo 2

2.1.4 Clasificadores

2.1.4.1 Red Neuronal Perceptrón Multicapa (MLP)

Una red de neuronas artificial puede pensarse como un grafo formado por neuronas organizadas por capas y relacionadas por conexiones que determinan la dirección del grafo (García and Osella,2003). En función de esta dirección, se pueden clasificar en:

- Redes de propagación hacia delante (*feedforward*): No tienen bucles. Algunos ejemplos pueden ser el perceptrón simple y el perceptrón multicapa.
- Redes recurrentes (*feedback*): Se producen bucles de retroalimentación.

Las redes neuronales poseen muchas ventajas:

- **Aprendizaje:** Tienen la capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.
- **Auto organización:** Crea su propia organización o representación de la información en su interior que recibe mediante una etapa de aprendizaje.
- **Tolerancia a fallos:** Si sufre algún daño puede seguir respondiendo de manera aceptable pues almacena la información de forma redundante.
- **Flexibilidad:** Puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada.
- **Tiempo real:** Por su estructura paralela se pueden obtener respuestas en tiempo real.

El perceptrón multicapa es una extensión del perceptrón simple. La topología está definida por un conjunto de capas ocultas, una capa de entrada y una de salida. No existen restricciones sobre la función de activación, aunque en general se suelen utilizar funciones sigmoideas (Juárez, 2018). Este modelo es el más utilizado en la actualidad. Se aplica en áreas como la codificación de información, traducción de texto en lenguaje hablado y Reconocimiento Óptico de Caracteres (OCR).

El algoritmo de retro propagación del error se basa en la regla del descenso del gradiente (Morera Munt, 2018). El funcionamiento de este algoritmo es versátil y potente. Dado un patrón de entrada que se aplica a la primera capa de neuronas de la red, se va propagando por las siguientes capas hasta que llega a la capa de salida, donde se compara la salida obtenida con la deseada. A continuación, se calcula el error para cada neurona de salida y éste es transmitido hacia atrás, por todas las capas intermedias, y se van modificando sus pesos sinápticos según dicho error y los valores de las salidas de las unidades precedentes ponderadas por sus pesos hasta

Capítulo 2

llegar a la capa de las unidades de entrada. Observar figura 15 en la sección de **anexos**.

2.1.4.2 Árbol de Decisión (Decisión Tree)

Como clasificador consiste en encontrar delimitadores dentro de los datos para ir tomando decisiones. Son modelos estadísticos que funcionan con bloques si-sino-entonces, donde si se cumple una cierta condición será más probable que se acabe clasificando una u otra categoría. Observar figura 16 en la sección de **anexos**.

Algunas ventajas de los árboles de decisión según (Scikit-learn developers, 2018) son:

- Simple de entender e interpretar. Los árboles se pueden visualizar.
- Requiere poca preparación de datos.
- Capaz de manejar datos tanto numéricos como categóricos y problemas de salida múltiple.
- Es posible validar un modelo utilizando pruebas estadísticas. Eso permite ver la fiabilidad del modelo.
- Funciona bien incluso si sus suposiciones se violan por el verdadero modelo a partir del cual se generaron los datos.

Algunas de las desventajas de los árboles de decisión según (Scikit-learn developers, 2018) es:

- El fenómeno de sobreajuste y la inestabilidad.

2.1.4.3 SVM (Support Vector Machine o máquinas de vectores de soporte).

Se trata de un método de clasificación basado en un entrenamiento supervisado (se conoce cuál es el resultado de cada muestra). A partir de un conjunto de muestras de entrenamiento se pueden etiquetar clases y entrenar un modelo SVM que prediga posteriormente a qué clase pertenece una nueva muestra. La principal ventaja frente a las redes neuronales es la gran eficiencia y rapidez del entrenamiento. Asimismo, otorga una alta precisión en la clasificación y por ello es tan utilizado en sistemas en tiempo real. Observar figura 17 en la sección de **anexos**.

2.1.4.4 Entrenamiento del clasificador

Esta fase consiste en entrenar el clasificador de tal manera, que a partir de un conjunto amplio de imágenes de las cuales se conoce su emoción, vaya aprendiendo qué características pertenecen a cada una de las emociones. Para ello toma cada imagen y realiza unos pasos parecidos a los descritos anteriormente: detecta la cara

Capítulo 2

de cada imagen, extrae sus características y asocia esas características a la emoción conocida que está expresando. Una vez que el clasificador ha sido entrenado el sistema ya está preparado para reconocer las emociones. Observar la figura 18 en la sección de **anexos**.

En OpenCV, que ya se había mencionado antes en el capítulo 1, existen algoritmos disponibles ya entrenados para detección facial o de objetos, algunos de estos son, LBP (Patrón Binario Local), Haar Cascade, de los cuales se obtuvo una breve descripción, este último se utilizó para reconocer la posición del rostro. También se menciona en el capítulo 1 las bibliotecas utilizadas, de las cuales es fundamental la utilización de *h5py (Facial Expression Model Weight)* que, integrada a Opencv facilita, en gran comparación a los algoritmos antes mencionados, la clasificación de las emociones y reconocerlas puesto que ya en su funcionamiento tiene los modelos para clasificar optimizando en tiempo y calidad de precisión al momento de dar la predicción. Por sus particularidades se concluyó que para esta herramienta el clasificador principal a utilizar es la propia biblioteca.

2.2 Levantamiento de requisitos

Son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requisitos reflejan la necesidad de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. Es el proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones (Somerville, 2011).

2.2.1 Requisitos funcionales

Los requisitos funcionales describen la interacción entre el sistema y su ambiente independientemente de su implementación (Somerville, 2011). El ambiente incluye al usuario y cualquier otro sistema externo que interactúa con el sistema.

Los requisitos funcionales del sistema son los siguientes:

- RF1: Activar cámara.
- RF2: Utilizar la cámara.
- RF3: Extraer rasgos del rostro con el método de extracción de características.
- RF4: Analizar las características del rostro de entrada y almacenar los datos.
- RF5: Reconocer la emoción en tiempo real.
- RF6: Clasificar la emoción detectada a partir del rostro de un usuario.
- RF7: Almacenar la emoción detectada.

Capítulo 2

- RF8: Generar interfaz de usuario principal.
- RF9: Generar interfaz de resultados.
- RF10: Cerrar aplicación.

2.2.2 Requisitos no funcionales

Los requisitos no funcionales describen aspectos del sistema que son visibles por el usuario que no incluyen una relación directa con el comportamiento funcional del sistema. Los requerimientos no funcionales incluyen restricciones como el tiempo de respuesta (desempeño), la precisión, recursos consumidos, seguridad, etc (Somerville,2011).

Requisitos no funcionales:

Usabilidad

- RNF1: El sistema debe presentar un acceso fácil y rápido, para facilitar el uso del mismo por usuarios con conocimientos básicos en el campo de la informática.
- RNF2: El sistema debe ser extensible, por lo que con el tiempo se le pueden adicionar nuevas funcionalidades.

Eficiencia

- RNF3: La aplicación debe de proporcionar la máxima precisión posible a la hora de reconocer la emoción.
- RNF4: La aplicación debe tener un tiempo de respuesta lo más corto posibles para que su funcionamiento sea óptimo en tiempo real.

Interfaz

- RNF5: Se debe mantener la misma distribución y diseño de los elementos del resto del sistema.
- RNF6: La aplicación debe tener una interfaz clara y sencilla.

Software

- RNF7: Intérprete de Python versión 3.8.
- RNF8: Framework Django 4.2
- RNF9: Utilizar como lenguaje Python en la versión 3.8
- RNF10: XAMPP versión 3.3.0

Hardware

- RNF11: PC Intel Dual Core 2.16 Hz o superior.

Capítulo 2

- RNF12: 4 GB de RAM o superior.
- RNF13: 10 GB de disco duro o superior.
- RNF14: Cámara de mínimo media resolución.

2.3 Historia de usuario

Las historias de usuario son la técnica utilizada para especificar los requisitos del software. Se trata de descripciones en las cuales se describe brevemente en lenguaje natural las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. El tratamiento de las historias de usuario es muy dinámico y flexible. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas. Se detalla que la estimación de puntos en de 0.10 delimita 1 día, 0.25 tres días aproximadamente y 0.50 una semana (Sánchez, 2016).

Las siguientes historias de usuario representadas son "Activar cámara " y "Utilizar cámara " pertenecientes a los requisitos funcionales 1 y 2 respectivamente. Las historias de usuarios correspondientes a los demás requisitos funcionales se podrán encontrar en la sección de **anexos**.

Tabla 4 Historia de usuario # 1.

Historia de usuario	
Número: 1	Nombre: Activar cámara.
Usuario: Usuario.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Media.
Puntos estimados: 0.25	Iteración asignada: 1
Programador responsable: Arisdalia Peña Román.	
Descripción: La aplicación debe mostrar un mensaje de permiso para activación de la cámara.	

Capítulo 2

Observaciones:

- Mostrar el siguiente mensaje de confirmación: “¿Desea activar la cámara?” y los botones “Aceptar” y “Cancelar”.
- La aplicación mostrará un mensaje con el siguiente texto: “No se detectó la cámara. Por favor verifique que la cámara esté conectada.” Y el botón “Aceptar”.
- El usuario presiona el botón cancelar. La aplicación mostrará un mensaje con el siguiente texto: “¿Está seguro que desea Cancelar?” Y el botón “Aceptar”.

Tabla 5 Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Utilizar la cámara.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Baja.
Puntos estimados: 0.25	Iteración asignada: 1
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema tiene que reconocer la cámara. El sistema utiliza la cámara y a través de ella muestra todo el análisis de la aplicación.	
Observaciones: La cámara debe estar activada para dar paso a la acción. La aplicación no analizará ni mostrará una cámara, debido que no la reconoce o no están cargados los drivers de la misma.	

2.4 Desarrollo del plan de iteraciones

Una vez definidas las historias se procede a la planificación de la etapa de implementación del sistema. Este plan, tiene como finalidad, mostrar la duración de cada iteración y el orden y prioridad de las historias de usuarios.

Capítulo 2

2.5 Plan de duración de las iteraciones

Tabla 14 Plan de duración de las iteraciones.

Iteración	Historia de Usuario	Puntos estimados (semanas)
1	RF1: Activar cámara.	4
	RF2: Utilizar la cámara.	
	RF3: Extraer rasgos del rostro con el método de extracción de características.	
2	RF4: Analizar las características del rostro de entrada y almacenar los datos.	5
	RF5: Reconocer la emoción en tiempo real.	
3	RF6: Clasificar la emoción detectada a partir del rostro de un usuario.	5
	RF7: Almacenar la emoción detectada.	
4	RF8: Generar interfaz de usuario principal.	4
	RF9: Generar interfaz de resultados.	
	RF10: Cerrar aplicación.	
Total		18

El desarrollo de la aplicación tomará un total 126 días dando paso así al desarrollo del plan de entregas. El plan de entregas puede variar o no en dependencia de las situaciones que se van dando en el transcurso del desarrollo con los recursos que estén o no en disposición para el avance de cada una de las funcionalidades necesarias de la herramienta y su correcto completamiento.

Capítulo 2

2.6 Plan de entregas

En el plan de entregas mostrado a continuación se especifican las fechas de inicio y de fin de cada iteración. Se tiene a consideración que si se trabaja corrido teniendo en cuenta los fines de semana este plan de entregas se extiende aproximadamente a 18 semanas en total.

Tabla 15 Plan de entregas

Iteraciones	Fecha de inicio	Fecha de fin
Iteración 1	29 de mayo de 2022	25 de junio de 2022
Iteración 2	26 de junio de 2022	17 de septiembre de 2022
Iteración 3	18 de septiembre de 2022	22 de octubre de 2022
Iteración 4	23 de octubre de 2022	19 de noviembre de 2022

2.7 Arquitectura de la solución

Según la organización de estándar IEEE la definición de arquitectura de software es:

“La arquitectura es la organización fundamental de un sistema encarnado en sus componentes, sus relaciones entre sí y con el entorno, y los principios que guían su diseño y evolución.”

Algunos ejemplos de arquitectura de software:

- Modelo Vista Controlador.
- Modelo Vista *Template*.
- En *pipeline*.
- Entre pares.
- En pizarra.
- Orientada a servicios (SOA del inglés *Service-Oriented Architecture*).
- Arquitectura de microservicios (MSA del inglés *MicroServices Architecture*). Algunos consideran que es una especialización de una forma de implementar SOA.

- Dirigida por eventos.
- Máquinas virtuales.
- Cliente -Servidor

2.7.1. Arquitectura de diseño a utilizar

En la arquitectura cliente/servidor los clientes (o programas que representan entidades que necesitan servicios) y los servidores (o programas que proporcionan servicios) son objetos separados desde un punto de vista lógico y que se comunican a través de una red de comunicaciones para realizar una o varias tareas de forma conjunta . Un cliente hace una petición de un servicio y recibe la respuesta a dicha petición; un servidor recibe y procesa la petición, y devuelve la respuesta solicitada (E. U. Informática en Segovia Departamento de Informática Universidad de Valladolid). De forma general esta arquitectura posee una estructura como la que se aprecia en la figura 19 en la sección de **anexos**.

Características de la arquitectura cliente/servidor (E. U. Informática en Segovia Departamento de Informática Universidad de Valladolid) :

- **Protocolos asimétricos:** hay una relación muchos a uno entre los clientes y un servidor. Los Clientes siempre inician un diálogo mediante la solicitud de un servicio. Los Servidores esperan pasivamente por las solicitudes de los clientes.
- **Encapsulación de servicios:** El servidor es un especialista, cuando se le entrega un mensaje solicitando un servicio, determina cómo conseguir hacer el trabajo. Los servidores se pueden actualizar sin afectar a los clientes en tanto que la interfaz pública de mensajes que se utilice por ambos lados, permanezca sin cambiar.
- **Integridad:** el código y los datos de un servidor se mantienen centralizados, lo que origina que el mantenimiento sea más barato y la protección de la integridad de datos compartidos. Al mismo tiempo, los clientes mantienen su independencia y “personalidad”.
- **Transparencia de localización:** el servidor es un proceso que puede residir en la misma máquina que el cliente o otra una máquina diferente de la red. El software cliente/servidor (middleware) habitualmente oculta la localización de un servidor a los clientes mediante la redirección de servicios. Un programa puede actuar tanto como cliente, como servidor o como cliente y servidor simultáneamente.

Capítulo 2

- **Intercambios basados en mensajes:** Los clientes y servidores son procesos débilmente acoplados que pueden intercambiar solicitudes de servicios y respuestas utilizando mensajes.
- **Modularidad, diseño extensible:** el diseño modular de una aplicación cliente/servidor permite que la aplicación sea tolerante a fallos. En sistemas tolerantes a fallos, los fallos pueden ocurrir sin causar la caída de la aplicación completa. En una aplicación cliente/servidor tolerante a fallos, uno o más servidores pueden fallar sin parar el sistema total mientras que los servicios proporcionados por los servidores caídos estén disponibles en otros servidores activos. Otra ventaja de la modularidad es que una aplicación cliente/servidor puede responder automáticamente al incremento o decremento de la carga del sistema mediante la incorporación o eliminación de uno o más servicios o servidores.
- **Independencia de la plataforma:** el software cliente/servidor “ideal” es independiente del hardware o sistemas operativos, permitiendo al programador mezclar plataformas de clientes y servidores. El entorno de explotación de clientes y servidores puede ser sobre diferentes plataformas, con el fin de optimizar el tipo de trabajo que cada uno desempeña.
- **Código reutilizable:** La implementación de un servicio puede utilizarse en varios servidores.
- **Escalabilidad:** Los sistemas cliente/servidor pueden ser escalados horizontal o verticalmente. El escalado horizontal significa añadir o eliminar estaciones clientes con un ligero impacto en el rendimiento. El escalado vertical significa la migración a una máquina servidora más grande y rápida o la incorporación de nuevas máquinas servidoras.
- **Separación de la funcionalidad del cliente/servidor:** El modelo cliente/servidor es una relación entre procesos que se ejecutan en la misma o en máquinas separadas. Un proceso servidor es un proveedor de servicios. Un cliente es un consumidor de servicios. El modelo cliente servidor proporciona una clara separación de funciones.
- **Recursos compartidos:** un servidor puede proporcionar servicios a muchos clientes al mismo tiempo, y regular el acceso de éstos a un conjunto de recursos compartidos.

2.7.2 Diagrama de componente

Para una correcta representación de la estructura de las clases que conforman el esqueleto de la herramienta se hace necesario el uso de un diagrama de componente. El diagrama de la figura 20 representará las conexiones entre las clases, además de los métodos que dan forma a la herramienta teniendo en evidencia el comportamiento de la arquitectura escogida.

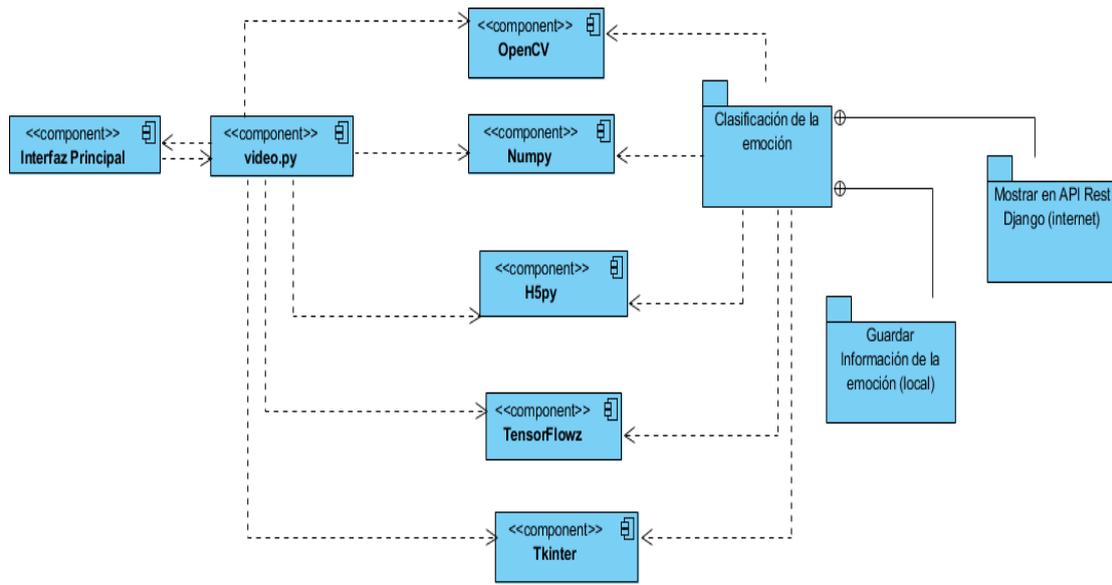


Figura 10 Diagrama de componente.

2.8 Tarjetas de Contenido, Responsabilidad y Colaboración

Las tarjetas de contenido, responsabilidad y colaboración o tarjetas CRC relacionan una clase con sus responsabilidades y los colaboradores que ayudan a realizar las mismas. La utilidad de estas radica en que facilita al equipo entender el proceso de diseño en una manera fácil y concisa. La tabla 16 muestra la tarjeta CRC correspondiente a la clase `video.py`. Las demás tarjetas CRC se podrán encontrar en la sección de los **anexos**.

Capítulo 2

Tabla 16 Tarjeta CRC # 1

Tarjeta CRC	
Clase: Video	
Responsabilidad	Colaboración
<ul style="list-style-type: none">• <code>search_face ()</code>• <code>search_emotion ()</code>	DeepFace RetinaFace

2.9 Conclusiones parciales

En este capítulo se describieron las características generales de la herramienta de reconocimiento facial de emociones. Se identificaron un total de 10 requisitos funcionales y 14 requisitos no funcionales. Mediante el uso de los RF, se redactaron las historias de usuarios, permitiendo posteriormente, la creación de un plan de 4 iteraciones y una estimación de 18 semanas aproximadamente para desarrollar la herramienta. Mediante la decisión del uso de la arquitectura Cliente Servidor se pudo lograr la confección de un correcto diagrama de componente. Con las clases ya conformadas y distribuidas adecuadamente, se posibilitó la descripción de las tarjetas CRC.

CAPÍTULO 3: IMPLEMENTACIÓN DE LA HERRAMIENTA DE RECONOCIMIENTO FACIAL Y PRUEBAS REALIZADAS.

En este apartado se detalla el progreso que se ha seguido para realizar la aplicación. Primero se describirán los primeros pasos realizados sobre Python, así como los primeros ejemplos realizados como introducción para la versión final de la aplicación. Así mismo, se comentarán los principales problemas encontrados al principio del desarrollo. Más adelante, partiendo de la estructura general de la aplicación se explicará en profundidad el procedimiento de cada uno de los pasos en el reconocimiento de una emoción. Aclarando las adaptaciones que se han tenido que realizar, así como las funciones de la librería usadas para ello. Así mismo, se mostrarán algunas pruebas realizadas durante el proceso.

3.1. Descripción y funcionamiento de la herramienta de reconocimiento facial de emociones.

A continuación, se muestra la guía de pasos para la obtención de la propuesta de solución:

1. El primer paso fue capturar los rostros en los que se reflejen las emociones: neutral, enojo, felicidad, sorpresa y tristeza, cada uno de ellos en una misma carpeta para su posterior interpretación de los datos. OpenCV ofrece clasificadores pre-entrenados no solos de rostros de personas, sino de ojos, sonrisa, caras de animales, entre otros.
2. Luego se procedió a utilizar el reconocedor con la siguiente librería: h5py y el método Haar Cascade este último integrado a OpenCV.
3. Finalmente se pasó a la fase de pruebas para probar el reconocimiento de cada una de las emociones detectadas.

3.1.2 Procesado de la imagen

El siguiente paso que se realizó fue la integración de la librería OpenCV para poder comunicar la cámara física con la aplicación para poder crear una aplicación propia. Esto facilita la posibilidad de crear una aplicación personalizada y que sea capaz de otorgar mayor rapidez y fluidez. Esta librería incluye algunos ejemplos para ello, desde una simple herramienta que muestra lo que se captura por la cámara hasta otras más complejas que son capaces de dar una primera aproximación para la detección de caras. Observar la figura 21 en la sección de **anexos**.

Capítulo 3

Muestra una simple aplicación que permite que la cámara del dispositivo detecte el rostro del usuario. Se puede observar un rectángulo de color rojo que muestra la detección del rostro del usuario.

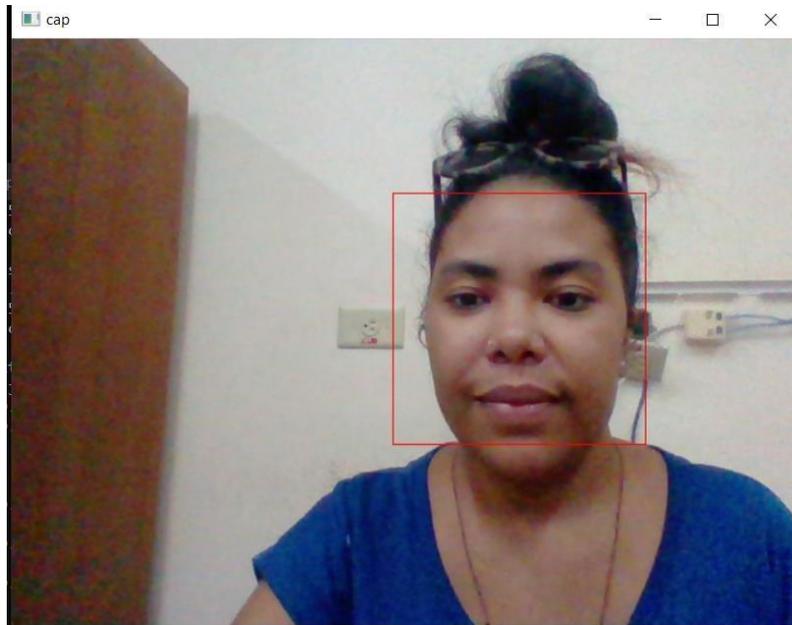


Figura 11 Detección del rostro.

3.1.3 Detección de la cara

En la primera etapa se realiza la detección de la cara. Para ello se pueden utilizar métodos como los expuestos en el Capítulo 1, por ejemplo, gracias a la librería OpenCV es posible realizar la detección de la posición de la cara con el algoritmo Haar Cascade.

El primero paso dentro de esta etapa consiste en que internamente mediante OpenCV se haga llamada al algoritmo. Una vez cargado el algoritmo, como paso previo para la detección se convierte la imagen a escala de grises en caso de que no lo esté. Después, para llevar a cabo la detección se pueden modificar una serie de parámetros que ayudan a limitar y ajustar la detección como:

- Factor de escala: parámetro para determinar cuánto se reduce la imagen para que pueda detectar la cara. A menor factor de escala mayor precisión, pero mayor retardo también.
- En este caso se emplea un factor de escala de 1,2, es decir, reducción del 12%.
- Número de vecinos: especifica el mínimo número de píxeles vecinos a tener en cuenta.

Capítulo 3

- A menor número menor precisión, pero más detecciones. En este caso se ha estimado que un vecino es el número óptimo.
- Tamaño mínimo y máximo: determinan cuales pueden ser el tamaño mínimo y máximo de la cara. En este caso se ha determinado que el tamaño mínimo tenga una altura al menos del 20% de la altura de la imagen.

3.1.4 Pre procesado

En este paso se tienen que procesar el video que segmenta cada toma como una imagen para analizar la emoción a detectar. En esta etapa del procesado de la imagen corresponde a la localización de las regiones principales de la cara: ojos, cejas, nariz y boca. El buen desempeño de un sistema de reconocimiento facial depende en gran parte de sus componentes que lo forman, existiendo 6 partes primordiales para que el sistema funcione adecuadamente. Observar la figura 23 en la sección de **anexos**.

- **Captación:** Es el proceso a través del cual se obtiene una imagen visual.
- **Pre procesamiento:** Incluye técnicas tales como la reducción de ruido y realce de detalles
- **Segmentación:** Es el proceso que divide a una imagen en objetos que sean de nuestro interés.
- **Descripción:** Es el proceso mediante el cual se obtienen características convenientes para diferenciar un tipo de objeto de otro, por ejemplo, tamaño y forma.
- **Reconocimiento:** Es el proceso que asocia un significado a un conjunto de objetos reconocidos.

Tras la importación de las librerías necesarias, como ocurre con la detección facial, OpenCV facilita una función cuyos parámetros se pueden modificar para realizar una búsqueda más precisa. Se aplica la misma configuración de los parámetros exceptuando el valor aplicado al tamaño mínimo y máximo a detectar. En este caso, el tamaño mínimo es 180x45(60% del ancho de la cara y 15% del alto) y el tamaño máximo 210x60 (70% del ancho de la cara y 20% del alto).

3.1.5 Localización de regiones y búsqueda de los puntos de interés.

Considerando que la mayor cantidad de información sobre la identidad de una persona está ubicada en la zona de los ojos y la boca, se busca mapear información en áreas coincidentes en dos rostros. Para esta estrategia las coordenadas de los ojos coinci-

Capítulo 3

den en los mismos puntos para todas las imágenes y con base en estas coordenadas se subdividen dos regiones, una para los ojos y otra exclusivamente para la boca. Entonces, cuando se desea analizar el rostro, se calcula la distancia mínima entre todos los puntos de la región de los ojos del sujeto desconocido, y de tal forma la aplicación internamente procesa la información de entrada proporcionada por los gestos del usuario.

De forma similar se calcula la distancia mínima entre los puntos para la zona de la boca. Posteriormente estas distancias son promediadas obteniendo la evaluación de similitud a una emoción determinada. Como se había mencionado anteriormente todo este proceso se hace internamente con la utilización de las librerías antes mencionadas y seleccionadas. Cuando se procede a ejecutar el proyecto queda reflejado los resultados de haber analizado un rostro y las emociones que son detectadas en tiempo real, este resultado queda guardado en un archivo dentro de la carpeta output del proyecto, en la figura 24 que aparece en la sección de los **anexos** se observa un ejemplo de los resultados guardados luego de análisis.

3.2 Tareas de ingeniería

La metodología XP, propone, por cada iteración, sean descritas las tareas de ingeniería en correspondencia a las historias de usuario de que derive. Cada tarea, describe desde el punto de vista del programador, los pasos para solucionar las necesidades planteadas en una historia de usuario; por lo que, una historia de usuario puede tener varias tareas de ingeniería asignadas. A continuación, se ejemplifican las tareas de ingeniería:

3.2.1 Iteración I

Dentro de la primera iteración se encuentran las funcionalidades de “cargar drivers de la cámara”, “conectar cámara al IDE pycharm” y “detectar rostros” y “mostrar rectángulo que indique la detección de un rostro” correspondientes a las historias de usuarios 1, 2 y 3. Se muestran las tablas correspondientes a las tareas 1 y 2 y en la sección de **anexos** las restantes tareas. Las tareas definidas para la iteración son:

- **Tarea N.1:** Cargar drivers de la cámara.
- **Tarea N.2:** Conectar cámara.
- **Tarea N.3:** Activar cámara.
- **Tarea N.4:** Detectar rostros.
- **Tarea N.5:** Mostrar rectángulo que indique la detección de un rostro.

Capítulo 3

Tabla 19 Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de historia de usuario: 2
Nombre de la tarea: Cargar drivers de la cámara.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.25
Fecha de inicio: 29 de mayo de 2022.	Fecha de fin: 1 de junio de 2022.
Programador responsable: Arisdalia Peña Román.	
Descripción: Cargar los drivers de la cámara en caso de que el dispositivo donde se despliegue el proyecto no los contenga.	

Tabla 20 Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de historia de usuario: 2
Nombre de la tarea: Conectar cámara.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.25
Fecha de inicio: 2 junio de 2022	Fecha de fin: 5 de junio de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Crear método que sea capaz, mediante la integración de la librería OpenCV, de conectar la cámara portable al IDE Pycharm, en caso que el dispositivo donde se despliegue el proyecto no contenga webcam.	

3.2.2 Iteración II

Dentro de la segunda iteración se encuentran las funcionalidades definidas como tarea N.6 y N. 7 correspondientes a las historias de usuario 4 y 5, las tablas correspondientes se muestran en la sección de los **anexos**. Las tareas definidas para la iteración son:

- **Tarea N.6:** Analizar las características del rostro de entrada y almacenar los datos.
- **Tarea N.7:** Detectar la emoción en tiempo real.

Capítulo 3

3.2.3 Iteración III

Dentro de la tercera iteración se encuentran las funcionalidades definidas como tareas N.8 y tarea N. 9 correspondientes a las historias de usuario 6 y 7, las tablas correspondientes se muestran en la sección de los **anexos**. Las tareas definidas para la iteración son:

- **Tarea N.8:** Definir las emociones en 5 clasificaciones (neutral, enojo, sorpresa, felicidad y tristeza).
- **Tarea N.9** Almacenar la emoción detectada.

3.2.4 Iteración IV

Dentro de la cuarta y última iteración se encuentran las funcionalidades definidas como tarea N.10, tarea N.11 y tarea N. 12 correspondientes a las historias de usuario 8, 9 y 10, las tablas correspondientes se muestran en la sección de los **anexos**. Las tareas definidas para la iteración son:

- **Tarea N.10:** Generar interfaz principal de usuario.
- **Tarea N.11:** Generar interfaz de resultados obtenidos.
- **Tarea N.12:** Cerrar la aplicación cuando se cierre el juego.

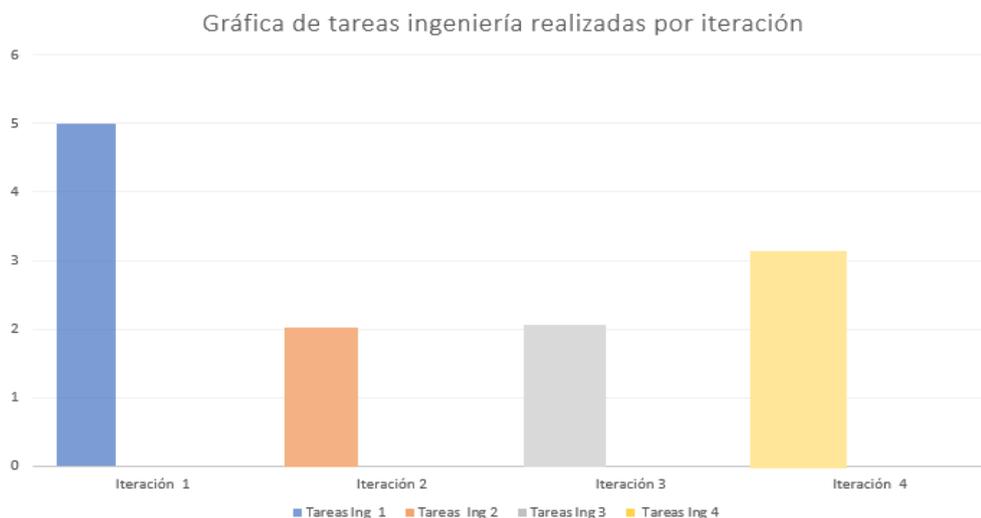
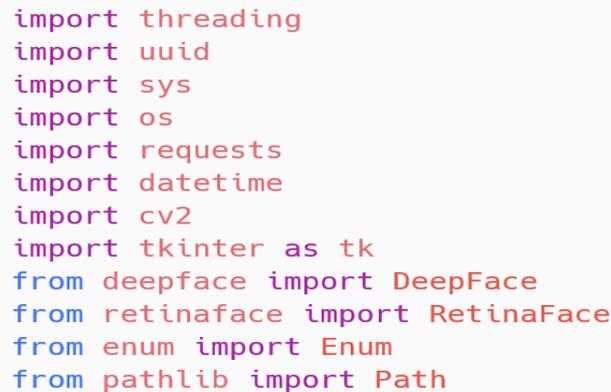


Figura 25 Gráfica de la tareas de ingeniería realizadas por iteración

3.3 Implementación

Para que la herramienta cumpliera con los estándares establecidos, así como un correcto funcionamiento de todos los requerimientos planteados, se necesitó utilizar un *script*. Dentro de sus líneas, están los métodos considerados los más importantes de la herramienta. Se seleccionaron partes del código del script lo cuales se muestran a continuación:

En el siguiente código se muestra la importación de las librerías más importantes para el correcto funcionamiento y análisis de la aplicación. De esta forma se asegura que una vez analizada la información se guarde.

A screenshot of a code editor window showing Python import statements. The window has three colored window control buttons (red, yellow, green) at the top left. The code is as follows:

```
import threading
import uuid
import sys
import os
import requests
import datetime
import cv2
import tkinter as tk
from deepface import DeepFace
from retinaface import RetinaFace
from enum import Enum
from pathlib import Path
```

Figura 26 Fragmento de código de la clase video.py

El siguiente fragmento de código que se muestra a continuación corresponde al método `search_face` donde en la primera parte del código representado es el encargado de la función de detección de la cara. En una segunda parte se refleja la funcionalidad de capturar fragmentos del video una vez abierta la cámara.

Capítulo 3

```
def search_face():
    face_cascade_name = cv2.data.haarcascades + 'haarcascade_frontalface_alt.xml'
    face_cascade = cv2.CascadeClassifier()
    if not face_cascade.load(cv2.samples.findFile(face_cascade_name)):
        print("Error loading xml file")

    cap = cv2.VideoCapture(0)
    c = 0
    state_label.config(text="Estado: {}".format(State.START.value))
    while cap.isOpened():
        _, frame = cap.read()
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        face = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)

        for x, y, w, h in face:
            cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 1)
```

Figura 27 Fragmento de código de la clase `viedo.py`

Otro método implementado es el método `search_emotion` el cual es el encargado de reflejar las emociones detectadas en tiempo real. También se puede observar la declaración de las emociones que se definieron y se tuvieron en cuenta para el análisis. Las emociones para poder ser reflejadas necesitan fundamentalmente de los gestos del rostro del usuario y su correcto posicionamiento delante de la cámara.

```
def search_emotion(file, emotion):
    faces = RetinaFace.extract_faces(file)
    for face in faces:
        obj = DeepFace.analyze(face, detector_backend='skip', actions=['emotion'])
        count_label.config(text="Count: {}".format(emotion))
        if(obj['dominant_emotion'] == 'fear'):
            emotion['fear'] = emotion['fear'] + 1
        elif(obj['dominant_emotion'] == 'angry'):
            emotion['angry'] = emotion['angry'] + 1
        elif(obj['dominant_emotion'] == 'neutral'):
            emotion['neutral'] = emotion['neutral'] + 1
        elif(obj['dominant_emotion'] == 'surprise'):
            emotion['surprise'] = emotion['surprise'] + 1
        elif(obj['dominant_emotion'] == 'sad'):
            emotion['sad'] = emotion['sad'] + 1
        elif(obj['dominant_emotion'] == 'happy'):
            emotion['happy'] = emotion['happy'] + 1
        # with open(output_file, 'a') as f:
        #     f.write(obj['dominant_emotion'] + '\n')
        #     f.close()
    #os.remove(file)
```

Figura 28 Fragmento de código de la clase `viedo.py`

Estos fragmentos de métodos antes expuestos junto a la importación de las librerías son los encargados del reconocimiento facial de emociones haciendo uso del funcionamiento de los aspectos estudiados teniendo en cuenta las particularidades de este

Capítulo 3

tipo de herramientas. Se puede concluir que el objetivo de esta aplicación parte fundamentalmente de estos algoritmos.

Otro de los métodos importantes es Gui y que en un breve momento trabaja de cara al usuario para obtener datos esenciales al inicio de activar la herramienta o una vez iniciada finalizar el proceso de análisis. Este algoritmo es el encargado de generar la interfaz principal y la interfaz encargada de mostrar los resultados obtenidos.

```
class Gui():
    def __init__(self):
        pass

    def start_record(self):
        reset_emotion()
        self.bindStart()
        hilo = threading.Thread(target=search_face, args=())
        hilo.start()

    def stop_record(self):
        self.bindStart()
        global break_loop
        break_loop = True

    def bindStart(self):
        start.bind("<Button>",
                  lambda e: self.start_record())
        stop.unbind("<Button>")

    def bindStop(self):
        stop.bind("<Button>", lambda e: self.stop_record())
        start.unbind("<Button>")

    def start(self):
        main_title.config(font=("Ariel", 18))
        main_title.place(x=80, y=0)
        user_name_label.place(x=15, y=40)
        user_name_data.place(x=130, y=40)
        game_name_label.place(x=15, y=70)
        game_name_data.place(x=130, y=70)
        state_label.place(x=17, y=200)
        start.place(x=140, y=100)
        stop.place(x=300, y=100)
        stop['state'] = 'disabled'
        state_label.config(text="Estado: {0}".format(State.INIT.value))
        self.bindStart()
        main_frame.pack(fill=tk.BOTH, expand=True)
        user_name_label.place(x=15, y=40)

        sencond_frame_label.place(x=0, y=0)
        angry_label.place(x=0, y=10)
        neutral_label.place(x=0, y=15)
        fear_label.place(x=0, y=20)
        surprise_label.place(x=0, y=25)
        sad_label.place(x=0, y=26)
        happy_label.place(x=0, y=27)
        close_btn.place(x=10, y=28)
        second_frame.pack()

        second_frame.pack(fill=tk.BOTH, expand=True)
        window.mainloop()

if __name__ == '__main__':
    gui = Gui()
    gui.start()
```

Figura 29 Código de la clase GUI

Capítulo 3

Otros aspectos fundamentales implementados están reflejados en los siguientes fragmentos de código. Estos están relacionados con datos complementarios como el contador de las emociones detectadas y la actualización de los datos recopilados. Esta actualización ocurre cada vez que se cambie el usuario y el nombre del videojuego. Otro aspecto de interés es el estado en el que se encuentre la herramienta, dígame un estado de inicio, de procesado de datos o finalización del proceso.

```

# Objecto de emociones
emotion = {
    "angry": 0,
    "neutral": 0,
    "fear": 0,
    "surprise": 0,
    "sad": 0,
    "happy": 0,
}

# Crear directorio temporal
if not os.path.exists(tmp_dir):
    os.mkdir(tmp_dir)

# Ver si la carpeta output existe
if not os.path.exists(os.path.join(Path(os.path.abspath(__file__)).parent, 'output')):
    os.mkdir(os.path.join(Path(os.path.abspath(__file__)).parent, 'output'))

# Estado de la aplicacion
class State(Enum):
    INIT = "Listo para capturar expresión"
    START = "Leyendo emociones"
    END = "Reconocimiento finalizado"

# Reiniciar el contador de emociones
def reset_emotion():
    global emotion
    emotion = {
        "angry": 0,
        "neutral": 0,
        "fear": 0,
        "surprise": 0,
        "sad": 0,
        "happy": 0,
    }

```

Figura 30 Código de aspectos fundamentales.

3.4 Diagramas de funcionamiento

3.4.1 Diagrama de clases

El diagrama de clases representado en la figura 31 muestra la integración de las clases fundamentales con la que se desarrolló la herramienta de reconocimiento facial

Capítulo 3

y la interacción que existe entre cada una de ellas. También se puede observar los métodos internos que están implementados expuestos anteriormente. Otro aspecto fundamental es la *Api Rest de Django* la cual almacena todos los datos correspondientes de una prueba para su posterior consulta o posible uso en otra aplicación.

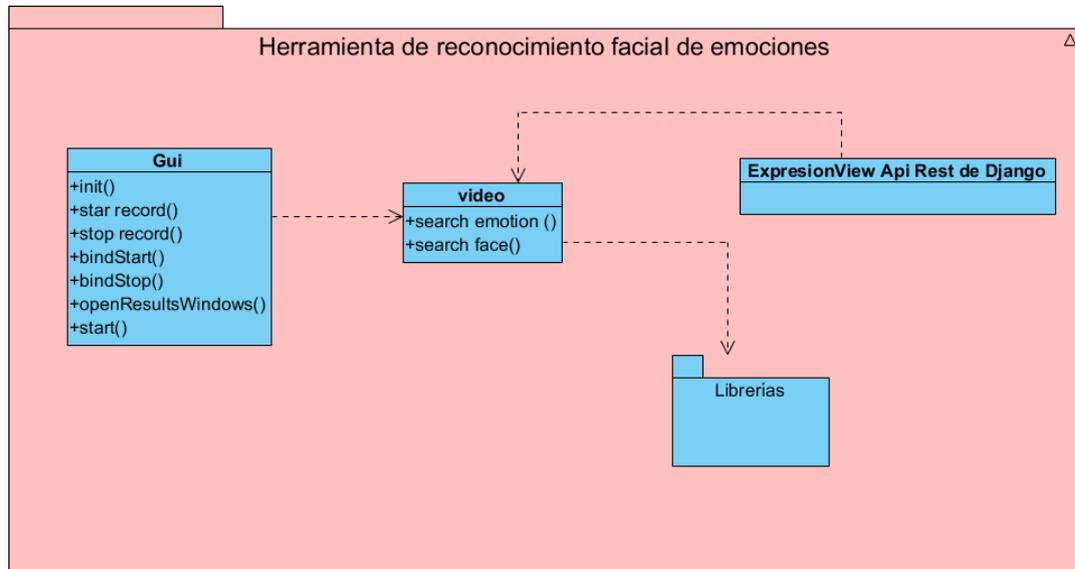
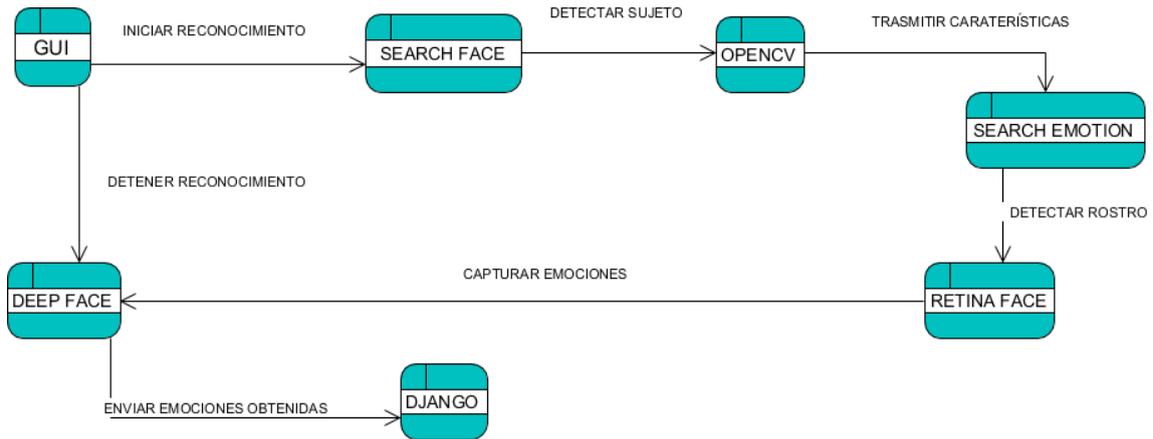


Figura 30 Diagrama de clases de la herramienta de reconocimiento facial.

3.4.2 Diagrama de flujo

El diagrama de flujo representado en la figura 31 muestra el comportamiento en cuanto a las funciones de cada uno de los procesos por los cuales se analizan las características del rostro del usuario a nivel global y concreto. La librería como numpy, h5py y tkinter son usadas por las librerías principales del sistema. Las dos primeras encuentran su propósito en el reconocimiento de rostro y de emociones, mientras que la última se encarga del maquetado visual del producto final.



3.5 Resultados obtenidos

Cuando se crea un producto informático es necesario una fase de validación la cual permite conocer que tanto el cliente está satisfecho con el resultado final. También es fundamental recoger los errores que aparecen en el ciclo de vida de dicho producto, de tal manera, darles una gestión apropiada para su corrección. A continuación, se muestra el resultado del producto final con varios usuarios y su interacción con el juego Coliseum donde se puede apreciar el reconocimiento de las emociones detectadas por la herramienta.



Figura 32 Prueba en Coliseum. Usuario Liany



Figura 33 Prueba en Coliseum. Usuario Dayana

3.6 Pruebas

Las pruebas de software son una parte integral del ciclo de vida del desarrollo de software. Las pruebas son la forma en que puede estar seguro acerca de la funcionalidad, el rendimiento y la experiencia del usuario. Ya sea que se realicen pruebas manualmente o a través de la automatización, cuanto antes y más a menudo se pueda llevar a cabo pruebas, más probable es que se identifiquen errores asegurando de que la aplicación de software haya sido revisada y auditada a fondo antes de que esté frente a los usuarios.

3.6.1 Pruebas de aceptación

La metodología XP propone las pruebas de aceptación, para comprobar el correcto funcionamiento de los requisitos planteados por el cliente. Las pruebas de aceptación, se realizan en cada una de las iteraciones; por lo que es necesaria la obtención de buenos resultados, para poder pasar a la siguiente iteración. A continuación, se muestran las pruebas realizadas.

Capítulo 3

3.6.1.1 Iteración I

Tabla 31 Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre: Activar cámara.	
Descripción: La aplicación debe reconocer la cámara y activar la cámara.	
Condiciones de ejecución: Una vez que el usuario decida abrir el videojuego la cámara debe ejecutarse.	
Pasos de ejecución: El usuario ejecuta el videojuego, luego la aplicación pide permiso de activar la cámara mientras el videojuego está en marcha.	
Resultados esperados: La aplicación muestra mensaje de permiso para que la cámara esté activada.	
No conformidades: No contiene los mensajes de permisos de activación para la cámara.	

Tabla 32 Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: 2
Nombre: Utilizar cámara.	
Descripción: La aplicación a través de la cámara debe de mostrar todo el análisis de la herramienta.	
Condiciones de ejecución: Una vez que el usuario decida abrir el videojuego la cámara de la aplicación debe ejecutarse.	
Pasos de ejecución: El usuario ejecuta el videojuego, luego la aplicación pide permiso de activar la cámara mientras el videojuego está en marcha y seguidamente la aplicación muestra la cámara en ejecución.	
Resultados esperados: Cámara en ejecución.	
No conformidades: No contiene los mensajes de permisos de activación para la cámara.	

Capítulo 3

Tabla 33 Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU5_P3	Historia de usuario: 5
Nombre: Detectar rostro.	
Descripción: La aplicación debe detectar el rostro del usuario.	
Condiciones de ejecución: Una vez que la cámara de la aplicación esté activada, a través de la misma, la aplicación deberá hacer reconocimiento de la presencia del rostro del usuario.	
Pasos de ejecución: La cámara debe tener el permiso de estar activada, luego de activarse, la aplicación procede hacer la detección del rostro del usuario.	
Resultados esperados: La aplicación detecta el rostro del usuario y muestra un cuadro color rojo acoplado al rostro.	
No conformidades: Ninguna.	

3.6.1.2 Iteración II

Tabla 34 Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU5_P4	Historia de usuario: 5
Nombre: Reconocer emoción.	
Descripción: La aplicación reconoce cada una de las emociones en tiempo real.	
Condiciones de ejecución: La aplicación debe tener la información necesaria para poder distinguir entre cada una de las emociones.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo mostrando o no una emoción.	
Resultados esperados: La aplicación distingue entre cada una de las emociones.	
No conformidades: Ninguna.	

Tabla 36 Prueba de aceptación # 5

Capítulo 3

Caso de prueba de aceptación	
Código: HU7_P5	Historia de usuario: 7
Nombre: Almacenar la emoción detectada.	
Descripción: La aplicación debe almacenar la información de las emociones detectadas.	
Condiciones de ejecución: La aplicación debe tener la información necesaria para poder distinguir entre un rostro o cualquier otro elemento.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo. La aplicación analiza los datos de entrada a partir del rostro del usuario reconoce cada una de las características de un rostro usando el método de extracción de características.	
Resultados esperados: La aplicación almacena cada una de las emociones que detectó de un usuario.	
No conformidad: Ninguna.	

3.6.1.3 Iteración III

Tabla 37 Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU6_P6	Historia de usuario: 6
Nombre: Estado neutral	
Descripción: La aplicación debe reconocer el estado neutral en tiempo real.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo sin reflejar ninguna emoción.	
Resultados esperados: La aplicación muestra el estado neutral detectado.	
No conformidad: Ninguna.	

Tabla 38 Prueba de aceptación # 7

Caso de prueba de aceptación

Capítulo 3

Código: HU6_P7	Historia de usuario: 6
Nombre: Estado enojo	
Descripción: La aplicación debe reconocer la emoción enojo en tiempo real.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo reflejando la emoción de enojo.	
Resultados esperados: La aplicación muestra el estado enojo detectado.	
No conformidad: Ninguna.	

Tabla 39 Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU6_P8	Historia de usuario: 6
Nombre: Estado sorpresa	
Descripción: La aplicación debe reconocer la emoción sorpresa en tiempo real.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo reflejando la emoción de sorpresa.	
Resultados esperados: La aplicación muestra el estado sorpresa detectado.	
No conformidad: Ninguna.	

Tabla 40 Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU6_P9	Historia de usuario: 6
Nombre: Estado alegría.	
Descripción: La aplicación debe reconocer la emoción alegría en tiempo real.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	

Capítulo 3

Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo reflejando la emoción de alegría.
Resultados esperados: La aplicación muestra el estado alegría detectado.
No conformidad: Ninguna.

Tabla 41 Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU6_P10	Historia de usuario: 6
Nombre: Estado tristeza.	
Descripción: La aplicación debe reconocer la emoción tristeza en tiempo real.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo reflejando la emoción de tristeza.	
Resultados esperados: La aplicación muestra el estado tristeza detectado.	
No conformidad: Ninguna.	

Tabla 42 Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU6_P11	Historia de usuario: 6
Nombre: Estado miedo	
Descripción: La aplicación debe reconocer la emoción miedo en tiempo real.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo reflejando la emoción de miedo.	
Resultados esperados:	

Capítulo 3

La aplicación muestra el estado miedo detectado.
No conformidad: Ninguna.

Tabla 43 Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU6_P12	Historia de usuario: 6
Nombre: Mostrar en pantalla emoción detectada en un bloc de notas.	
Descripción: La aplicación debe mostrar la emoción en tiempo real a través de un texto.	
Condiciones de ejecución: La aplicación debe estar en ejecución a la par del videojuego.	
Pasos de ejecución: El usuario debe estar mirando de forma frontal hacia la pantalla del dispositivo reflejando alguna emoción.	
Resultados esperados: La aplicación muestra de forma textual en pantalla la emoción que detectó.	
No conformidad: Solo muestra el resultado de las emociones de un usuario.	

3.6.1.4 Iteración IV

Tabla 44 Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU8_P13	Historia de usuario: 8
Nombre: Generar interfaz principal de usuario.	
Descripción: Crear un algoritmo donde se defina un visual de una interfaz principal donde se encuentren los campos a insertar de: <ul style="list-style-type: none">• Nombre del usuario.• Nombre del juego. Se encuentra en la parte inferior de la interfaz los botones de: <ul style="list-style-type: none">• Iniciar• Terminar	
Condiciones de ejecución: El usuario debe de llenar los campos solicitados.	
Pasos de ejecución: Una vez rellenos los campos solicitados, el usuario debe	

Capítulo 3

pulsar el botón "Iniciar "
Resultados esperados: La aplicación cambia al estado de listo para empezar el proceso de análisis.
No conformidad: Cuando no se introduce alguno de los campos o ninguno y se presiona el botón de "Iniciar " no se arroja la excepción del mensaje correspondiente a la introducción de los campos obligatorios.

Tabla 45 Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU9_P14	Historia de usuario: 9
Nombre: Generar interfaz de resultados obtenidos.	
<ul style="list-style-type: none">• Descripción: Muestra un listado de las emociones detectadas y la cantidad de veces que es encontrada la misma. También muestra al final de la ventana el botón de "Cerrar".	
Condiciones de ejecución: El usuario debe de llenar los campos solicitados.	
Pasos de ejecución: Una vez rellenos los campos solicitados, el usuario debe pulsar el botón "Iniciar " y el análisis pasar al estado de listo para comenzar el proceso.	
Resultados esperados: La aplicación cambia al estado de procesado del análisis para mostrar los resultados obtenidos.	
No conformidad: Cuando no se introduce alguno de los campos o ninguno y se presiona el botón de "Iniciar " no se arroja la excepción del mensaje correspondiente a la introducción de los campos obligatorios.	

Tabla 46 Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU10_P15	Historia de usuario: 10
Nombre: Cerrar aplicación.	
Descripción: La aplicación debe cerrarse correctamente.	
Condiciones de ejecución: El usuario debe cerrar el videojuego.	

Capítulo 3

Pasos de ejecución: Cuando el usuario decida cerrar el videojuego la aplicación detendrá todo proceso y también se cerrará.

Resultados esperados:

La aplicación se cierra correctamente.

No conformidad: La aplicación solo cierra si el usuario presiona el botón de cerrar cámara.

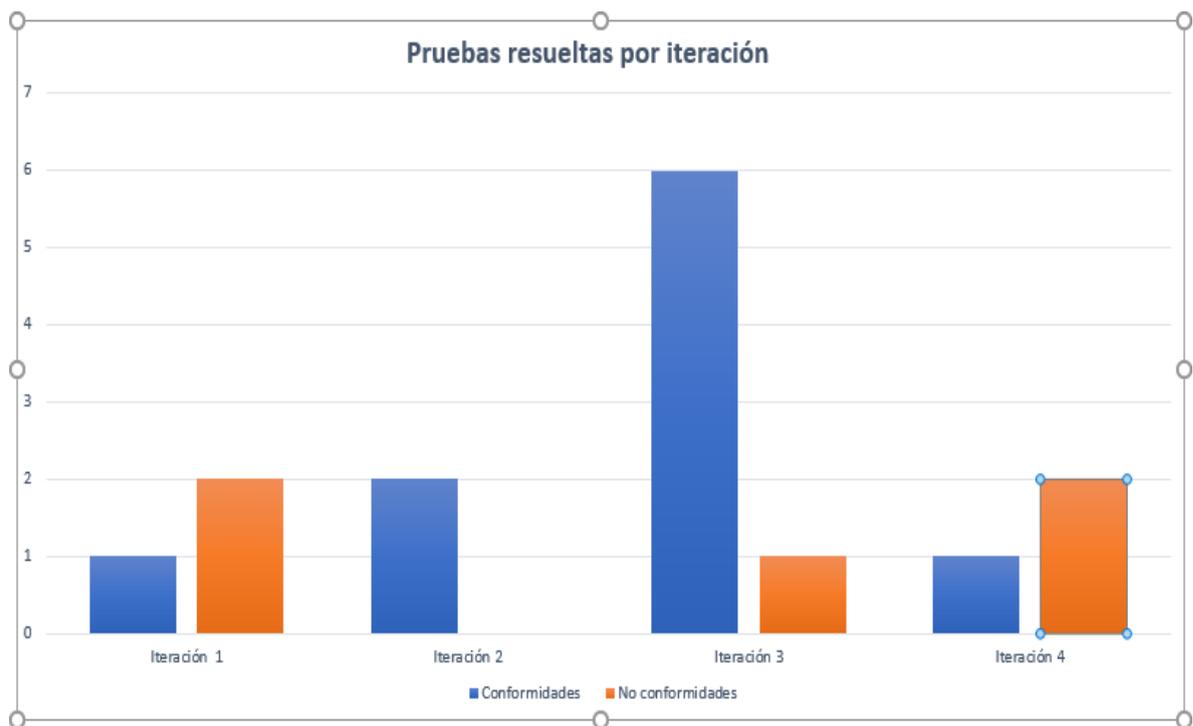


Figura 34 Gráfica de conformidades y no conformidades por iteración .

3.7 Conclusiones parciales

En este capítulo se abordaron los temas principales de la implementación de la herramienta y las pruebas de aceptación realizadas. En primera instancia se describieron un total de 12 tareas de ingeniería, planteando las acciones a seguir por el equipo de programación para conformar la herramienta de reconocimiento facial de emociones. Una vez terminada la fase de prueba se arriba a que se realizaron 15 pruebas de aceptación, siendo 10 satisfactorias, y 5 no satisfactorias, demostrando el cumplimiento de casi todos los objetivos propuestos.

Conclusiones

Conclusiones.

En este trabajo se definió como objetivo inicial realizar una propuesta de arquitectura orientada a la detección de emociones para así tener un nivel de calidad con los videojuegos desarrollados en el Centro de Tecnologías Interactivas. Teniendo en cuenta el estado del arte actual no es posible construir un sistema fiable para cuantificar hasta qué punto esta propuesta de reconocimiento facial de emociones puede mejorar la calidad del juego en cuanto a la satisfacción del usuario. Se tuvieron varios aspectos en cuenta en la realización de este trabajo:

- El rostro es el principal sistema de señales para mostrar las emociones.
- Se estudiaron las emociones como elemento fundamental utilizadas en la mayoría de los sistemas de detección automática de emociones en la actualidad.
- Las expresiones faciales de emociones resultan importantes en la interacción entre los humanos.
- Se hace referencia a las herramientas, métodos y sistemas homólogos para reconocer, interpretar, procesar y generar emociones humanas mejorando la interacción entre el usuario y la computadora.

Recomendaciones

Recomendaciones.

- Se deben mejorar los parámetros de la luz y aditamentos que se puedan encontrar en el análisis.
- Se debe tener en cuenta todos los parámetros biométricos que deben tratarse cuidadosamente según la política de privacidad y ética de no violentar los datos del usuario. Estos datos no deben difundirse y el usuario debe ser informado de los fines para los cuales se tratarán los mismos.
- La ubicación del sistema se desplegará teniendo en cuenta el lugar, donde el posicionamiento del sensor será estratégico para tener mejor calidad del procesado de los datos.
- La herramienta está diseñada para ser lo menos intrusiva posible y no interfiera en el proceso del juego, así, de esta manera se pueda obtener los resultados del análisis en períodos de tiempo cortos.
- La distancia entre el dispositivo y el usuario debe de ser relativamente corta para mejor obtención de los datos.
- Se arriba entonces luego de esta breve investigación que se ha podido construir una herramienta de reconocimiento facial de emociones con las particularidades necesarias para su debido funcionamiento.

Bibliografía

Referencias Bibliográficas

- Lionel R. Baquero Hernández, Dayana Mendoza Peña , Osviel Rodríguez Valdés , Omar Mar Cornelio 1 1 Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso.Serie Científica de la Universidad de las Ciencias Informáticas Vol. 9, No. 7, julio, 2016 ISSN: 2306-2495 | RNPS: 2343 Pág. 7-20 <http://publicaciones.uci.cu>
- Luis Blázquez Pérez ,Reconocimiento Facial Basado en Puntos Característicos de la Cara en entornos no controlados, Escuela Politécnica Superior Universidad Autónoma de Madrid enero 2013.
- P. Ekman, W.V. Friesen, “Constants across Cultures in the face and emotion. “Journal of Personality and Social Psychology, vol.17, no. 2, pp. 124–129, 1971
- P. Ekman, “Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique,” Psychological Bulletin, vol.115, no. 2, pp. 268-287, Mar.1994
- P. Viola y M. Jones, “Rapid object detection using a boosted cascade of simple features,” Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001, vol. 1, págs. I–511–I–518,2001.
- P. Viola and M.J. Jones, “Robust real-time object detection,” Int. Journal of Computer Vision, vol.57, no. 2, pp. 137-154, Dec. 2004.
- A. Harvey, Interviewee, Adam Harvey Explains Viola-Jones Face Detection. [Entrevista].2012
- Timo Ojala, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” Pattern Analysis and Machine Intelligence, IEEE Transactions on vol.24, no.7, págs.971–987, 2002.
- E. Hjelm, B. Kee Low, “Face Detection: A Survey,”Computer Vision and Image Understanding, 83, 236-274 April 2001.
- Diego, Isaac Martín de; Serrano, Ángel; Conde, Cristina; Cabello, Enrique
- TÉCNICAS DE RECONOCIMIENTO AUTOMÁTICO DE EMOCIONES Teoría de la Educación. Educación y Cultura en la Sociedad de la Información, vol. 7, núm. 2, diciembre, 2006, pp. 107-127
- Hernández Domínguez, Antonio; Baluja García, Walter, Principales mecanismos para el enfrentamiento al phishing en las redes de datos, Revista Cubana de Ciencias Informáticas, vol. 15, Esp., 2021, Octubre-Diciembre, pp. 413-441, Editorial Ediciones Futuro.

Bibliografía

- WILLIAM A. CASTRILLON Ingeniero Electrónico. Investigador Universidad Tecnológica de Pereira william.castrillon@gmail.com DAMIAN A. ALVAREZ Ingeniero Electricista. Profesor Catedrático Universidad Tecnológica de Pereira damianalvarez@ohm.utp.edu.co ANDRÉS F. LÓPEZ Ingeniero Electricista. Profesor catedrático universidad tecnológica de técnicas de extracción de características en imágenes para el reconocimiento de expresiones faciales. scientia et technica año xiv, no 38, junio de 2008. universidad tecnológica de pereira. issn 0122-1701.
- El grupo de protección de las personas en lo que respecta al tratamiento de datos personales creado por la Directiva 95/46/CE del Parlamento Europeo y del Consejo, de 24 de octubre de 1995, Dictamen 3/2012 sobre la evolución de las tecnologías biométricas, adoptado el 27 de abril de 2012.
- Miguel Ángel Vázquez López. Sistema de Reconocimiento Facial Mediante Técnicas de Visión Tridimensional, marzo de 2014.
- José Augusto CADENA MOREANO. Técnica eficiente para reconocimiento facial global utilizando wavelets y máquinas de vectores de soporte en imágenes 3D, Lima, Perú 2021.
- Mario Piattini .Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión. Madrid: Editorial Rama (1996).
- Turban, E. (1995). Decision Support and Expert Systems (4ta edición). EE.UU. Prentice-Hall.
- Ian Somerville, Ingeniería del Software, 2011.
- Esteban Andrés García, Germán Leandro Osella Massa, Evolución de Redes Neuronales mediante Sistemas de Reescritura, diciembre 2003.
- Maida, EG, Pacienza, J. Metodologías de desarrollo de software [en línea]. Tesis de Licenciatura en Sistemas y Computación. Facultad de Química e Ingeniería "Fray Rogelio Bacon". Universidad Católica Argentina, 2015. Disponible en: <http://bibliotecadigital.uca.edu.ar/repositorio/tesis/metodologias-desarrollo-software.pdf>
- Marcelino Rodríguez Cancio, Biblioteca de componentes Flash para aplicaciones educativas multimedia Santa Clara, 2008.
- Daniel Martín Martínez, Aplicación de modelos de aprendizaje automático en microcontroladores, 2021
- Yamila Ortuño Sánchez , Herramienta informática para la identificación de la ambigüedad en textos de la legislación cubana, Junio 2016.

Bibliografía

- Cristian Domínguez Gómez, Aplicación práctica de machine learning sobre imágenes para un contexto de seguridad, julio 2018.
- Diego Zapatero Olmedillo, Herramienta de reconocimiento facial de emociones en Android, junio 2016.
- Softekk Company, Next Gen Software : Languages and Tools,julio 2018.
- EL MODELO CLIENTE/SERVIDOR, documento E. U. Informática en Segovia Departamento de Informática Universidad de Valladolid

Anexos

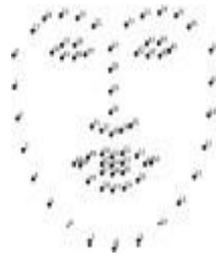


Figura 12. Representación de 68 marcas faciales en el rostro humano.

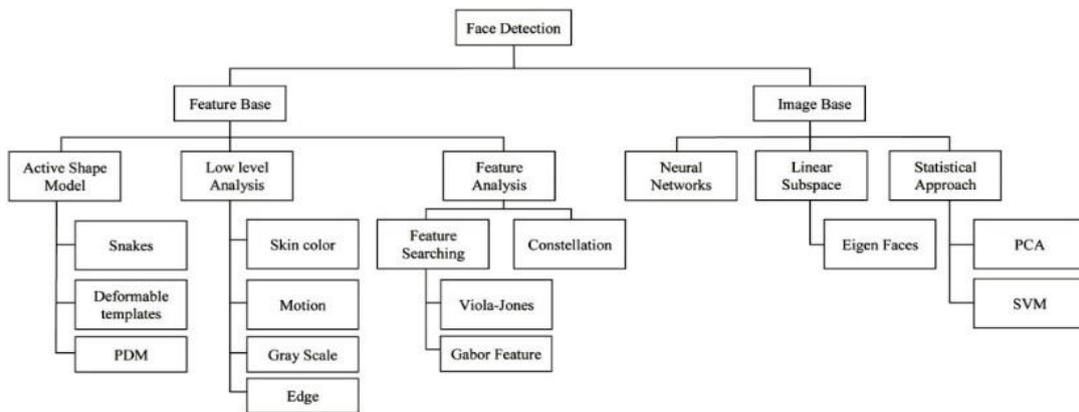


Figura 13 Métodos de detección facial. Fuente Zapatero,2016

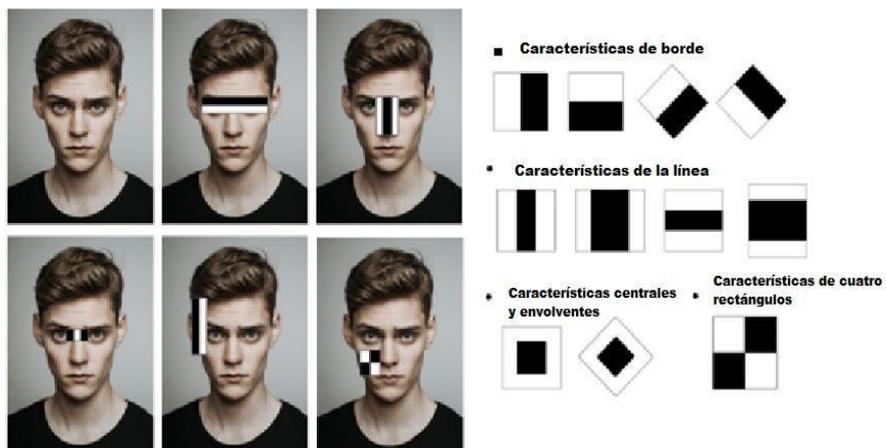


Figura 14 Algoritmo Viola-Jones.

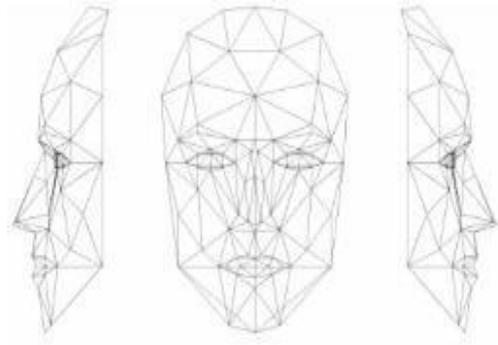


Figura 14 Máscara Candide.

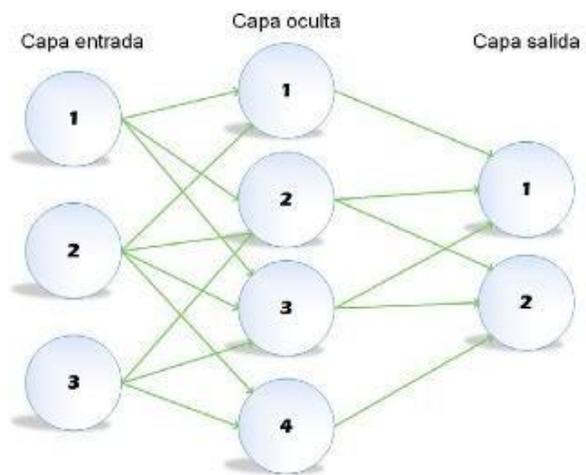


Figura 15 Sistema de red neuronal.

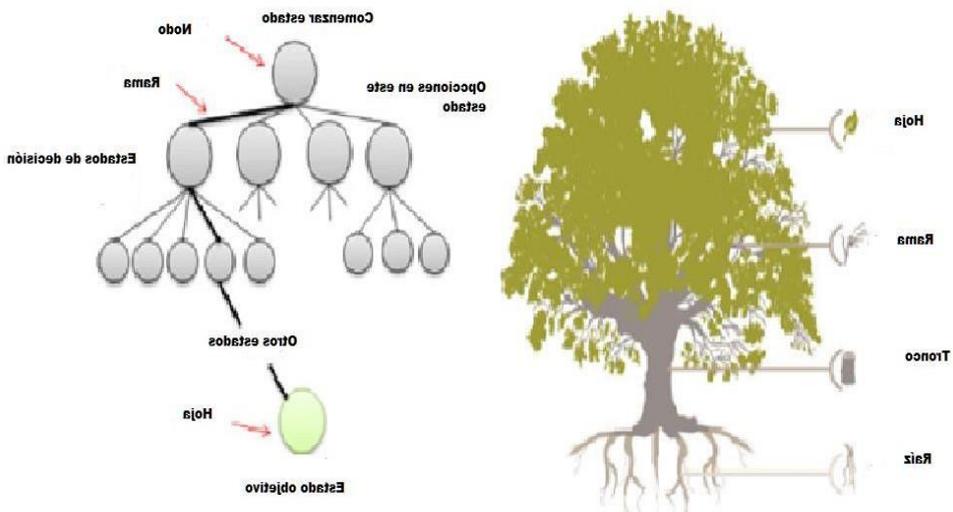


Figura 16 Estructura de árbol de decisión.



a)

b) Hiperplano óptimo o canónico

Figura 17 Ejemplo de hiperplanos SVM.



Figura 18 Diagrama de bloques reconocimiento facial de emociones.

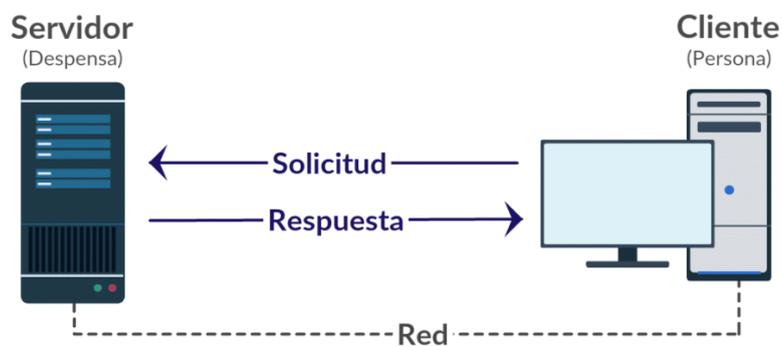


Figura 15 Ciclo de vida de la arquitectura cliente servidor .

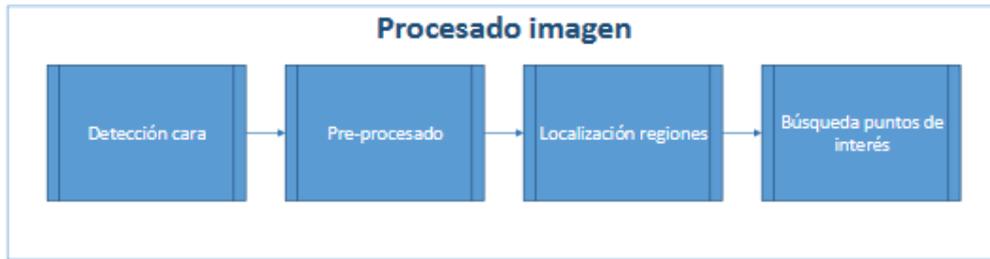


Figura 16 Etapas proceso de la imagen.

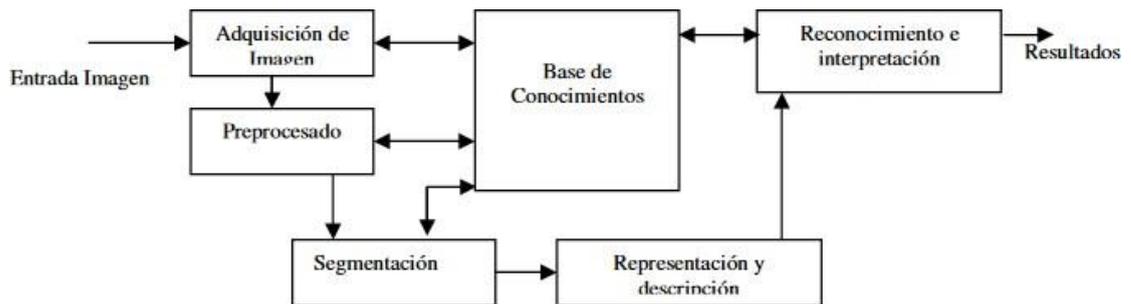


Figura 17 Etapas fundamentales para el pre procesamiento de una imagen.

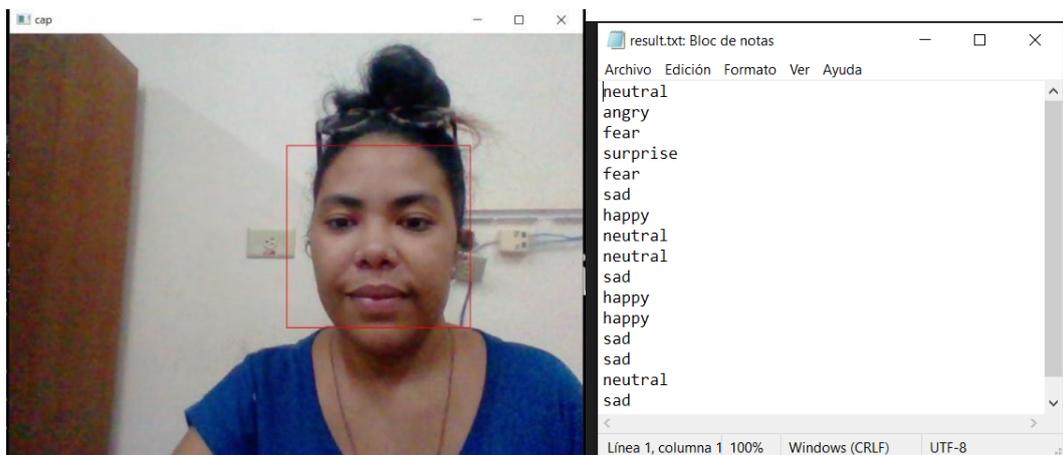


Figura 18 Resultados obtenidos luego del análisis del rostro.

Tabla 6 Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Extraer rasgos del rostro con el método de extracción de características.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 1.50	Iteración asignada: 1
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema detecta un rostro y extrae las características del mismo para poder reconocer una emoción.	
Observaciones: <ul style="list-style-type: none"> • El rostro del usuario debe estar en correcto posicionamiento para que el sistema haga el análisis y seguidamente extraiga sus características. • La aplicación muestra un cuadro color verde con la altura y el ancho adecuados al rostro del usuario. • El rostro del usuario no está bien posicionado, la aplicación no analiza el rostro y por tanto no puede extraer las características. 	

Tabla 7 Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Analizar las características del rostro de entrada y almacenar los datos.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta
Puntos estimados: 1.25	Iteración asignada: 2
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema tiene que analizar internamente las características del rostro	

Anexos

de entrada y almacenar los datos obtenidos.

Tabla 8 Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Reconocer la emoción en tiempo real.
Usuario: Usuario.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta.
Puntos estimados: 1.25	Iteración asignada: 2
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema tiene que analizar el rostro del usuario para luego detectar una emoción y clasificarla como una de las siguientes, ira, asco, miedo, alegría, sorpresa o simplemente se mantiene en neutral.	
Observaciones: <ul style="list-style-type: none">• La cámara debe estar activada para dar paso a la acción.• La aplicación no analizará ni mostrará una emoción debida que no reconoce la combinación de las unidades de acción analizadas.	

Tabla 9 Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Clasificar la emoción detectada a partir del rostro de un usuario.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Alta
Puntos estimados: 1.25	Iteración asignada: 3
Programador responsable: Arisdalia Peña Román.	
Descripción: La herramienta se pone en marcha teniendo en cuenta la privacidad del	

Anexos

usuario, luego hace un análisis del rostro del usuario extrayendo sus características para luego detectar una emoción y clasificarla como una de las siguientes, ira, miedo, alegría, sorpresa o simplemente se mantiene en neutral.

Observaciones:

- El usuario da permiso a la herramienta para obtener las características de su rostro.
- Se detectó un rostro y se clasificó la emoción.
- La aplicación no analizará ni mostrará una emoción debido que la posición del rostro de la persona no está en posición frontal.

Tabla 10 Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Almacenar la emoción detectada.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Baja.
Puntos estimados: 1.25	Iteración asignada: 3
Programador responsable: Arisdalia Peña Román.	
Descripción: La aplicación debe almacenar la emoción detectada.	
Observaciones: La aplicación no analiza correctamente los datos de entrenada y no puede reconocer la emoción, por tanto, falla en su almacenamiento o da un mal pronóstico.	

Tabla # 11 Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Generar interfaz de usuario principal.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Baja

Anexos

Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema debe mostrar una una interfaz principal donde el usuario pueda introducir su nombre y el nombre del juego.Luego aparecen más abajo los botones de iniciar y terminar para dar comienzo o finalizar el proceso.	
Observaciones: Aparece un mensaje de estado: "Listo para capturar la expresión" luego que se hayan introducido los datos de inicio.	

Tabla # 12 Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Generar interfaz de resultados.
Usuario: Sistema.	
Prioridad en negocio: Alta.	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema debe mostrar en una interfaz los resultados obtenidos luego del análisis de la herramienta con las características del usuario.	
Observaciones: Se muestra un listado donde aparecen las emociones detectadas y la cantidad de veces que estas aparecen, más abajo aparece el botón "Cerrar".	

Tabla 13 Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Cerrar aplicación.
Usuario: Usuario.	

Anexos

Prioridad en negocio: Alta.	Riesgo en desarrollo: Baja.
Puntos estimados: 1	Iteración asignada: 4
Programador responsable: Arisdalia Peña Román.	
Descripción: El sistema está en funcionamiento y solo dejará de funcionar mientras el usuario le haya dado pausa a la partida del videojuego o este desee cerrar el mismo.	
Observaciones: <ul style="list-style-type: none"> • La aplicación se detuvo e hizo un fallo de funcionamiento sin que el usuario interactuara. • Debe mostrar un mensaje de fallo de funcionamiento que diga lo siguiente: “Fallo en la aplicación” 	

Tabla 17 Tarjeta CRC # 2

Tarjeta CRC	
Clase: ExpresionView	
Responsabilidad	Colaboración
	LisMixin CreateMixin
Nota: Esta tarjeta CRC no tiene métodos debido que Django rest framework trae mecanismos para automatizar las acciones básicas sobre el modelo.	

Tabla 18 Tarjeta CRC # 3

Tarjeta CRC	
Clase: Gui	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • init () • star record () • stop record () • bindStart () 	Search face

Anexos

<ul style="list-style-type: none">• bindStop ()• openResultsWindows ()• start ()	
---	--

Tabla 11 Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de historia de usuario: 1
Nombre de la tarea: Activar cámara.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Fecha de inicio: 6 junio de 2022	Fecha de fin: 12 de junio de 2022
Programador responsable: Arisdalia Peña Román	
Descripción: Crear método que sea capaz de mostrar los mensajes de permisos de activación de cámara y mediante la integración de la librería OpenCV activarse de forma correcta.	

Tabla 22 Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de historia de usuario: 3
Nombre de la tarea: Detectar rostros.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.50
Fecha de inicio: 13 de junio de 2022	Fecha de fin: 19 de junio de 2022
Programador responsable: Arisdalia Peña Román	
Descripción: Cargar método q sea capaz, mediante la integración de la librería opencv, hacer la clasificación para la detección de rostros.	

Tabla 23 Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de historia de usuario: 3
Nombre de la tarea: Mostrar rectángulo que indique la detección de un rostro.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.50

Anexos

Fecha de inicio: 19 de junio de 2022	Fecha de fin: 25 de junio de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Utilizar método que dibuje un rectángulo indicando que se ha detectado un rostro.	
No conformidades: Ninguna.	

Tabla 24 Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de historia de usuario: 4
Nombre de la tarea: Analizar las características del rostro de entrada y almacenar los datos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1.25
Fecha de inicio: 25 de junio de 2022	Fecha de fin: 13 de julio de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Utilizar un algoritmo que sea capaz, mediante la integración de la librería OpenCV, de realizar una predicción correcta de la emoción partir de los datos de entrada analizados para luego almacenarlos.	

Tabla 25 Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de historia de usuario: 5
Nombre de la tarea: Detectar la emoción en tiempo real.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1.25
Fecha de inicio: 14 de julio de 2022	Fecha de fin: 17 de septiembre de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Utilizar un algoritmo que sea capaz de realizar una predicción correcta de la emoción en tiempo real mediante los datos de entrada.	
Nota: En esta tarea de ingeniería se comenzó el día 14 de julio a desarrollarse, pero debido a las vacaciones planificadas correspondientes al equipo de trabajo se retomó el desarrollo de la misma el día 1 de septiembre concluyendo el día 17 del mismo mes.	

Anexos

Tabla 26 Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de historia de usuario: 6
Nombre de la tarea: Definir las emociones en 6 clasificaciones (neutral, enojo, miedo, sorpresa, felicidad y tristeza).	
Tipo de tarea: Desarrollo	Puntos estimados: 1.25
Fecha de inicio: 18 de septiembre de 2022	Fecha de fin: 5 de octubre de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Crear un algoritmo donde se definan las 5 emociones a tratar.	

Tabla 27 Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de historia de usuario: 7
Nombre de la tarea: Almacenar la emoción detectada.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1.25
Fecha de inicio: 6 de octubre de 2022	Fecha de fin: 22 de octubre de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Utilizar un algoritmo de forma tal que se almacene la emoción detectada y se muestre esa información.	

Tabla 28 Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de historia de usuario: 8
Nombre de la tarea: Generar interfaz principal de usuario.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha de inicio: 23 de octubre de 2022	Fecha de fin: 30 de octubre de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Crear un algoritmo donde se defina un visual de una interfaz principal donde se encuentren los campos a insertar de: <ul style="list-style-type: none">• Nombre del usuario.• Nombre del juego.	

Anexos

Se encuentra en la parte inferior de la interfaz los botones de:

- Iniciar
- Terminar

No conformidades: Cuando no se introduce alguno de los campos o ninguno y se presiona el botón de "Iniciar" no se arroja la excepción del mensaje correspondiente a la introducción de los campos obligatorios.

Tabla 29 Tarea de ingeniería # 11

Tarea	
Número de tarea: 12	Número de historia de usuario: 10
Nombre de la tarea: Generar interfaz de resultados obtenidos.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.50
Fecha de inicio: 1 de noviembre de 2022	Fecha de fin: 7 de noviembre de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Muestra un listado de las emociones detectadas y la cantidad de veces que es encontrada la misma. También muestra al final de la ventana el botón de "Cerrar".	
No conformidades: Ninguna.	

Tabla 30 Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de historia de usuario: 10
Nombre de la tarea: Cerrar la aplicación cuando se cierre el juego.	
Tipo de tarea: Desarrollo.	Puntos estimados: 0.50
Fecha de inicio: 8 de noviembre de 2022	Fecha de fin: 15 de noviembre de 2022
Programador responsable: Arisdalia Peña Román.	
Descripción: Una vez el usuario cierre el videojuego la herramienta se cerrará protegiendo así la información del usuario.	