



**Título:** Clasificación automática de emociones en las personas  
utilizando el dispositivo Kinect.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE  
INGENIERO EN CIENCIAS INFORMÁTICAS

**Autor:** Yaimara Fleites Reynaldo.

**Tutor:**

Ing. Andy Trujillo Rivero.

**Co-Tutores:**

MSc. Susej Beovides Luis.

Ing. Ernesto A. Leyva Piñeda.

Ing. Esmaykel Vázquez Ávila.

La Habana, 2016

*“La vida es bella...”*

*Dedicatoria:*

*A mi abuela (Mau), donde estés quiero que sepas que te llevo muy dentro y cada paso que doy en la vida está inspirado en tu recuerdo.*

*A mi abuelo (Papi), por esa paciencia infinita. Al amor de los dos porque es lo más genuino y bello que he conocido. Gracias por ser mis abuelos, siempre los amaré.*

*A mi mamá y mi papá por todo el amor y apoyo que siempre me han brindado, porque todo lo que soy y llegaré a ser es gracias a ustedes.*

*A mi hermanita del alma y a mis bellos sobrinos, ojalá yo sea un ejemplo para ellos como tú lo has sido para mí.*

*A todos ellos por ser lo más hermoso que tengo en la vida.*

*Agradecimientos.*

*Le agradezco a mi madre, por enseñarme a defender las cosas en las que creo por encima de todo, por inculcarme el amor a esta tierra nuestra y por ser el ejemplo de persona más perfecto que he conocido.*

*A mi padre por enseñarme el valor de la familia, por ser un padre antes que todo y velar junto a mamá por nuestro futuro.*

*Les agradezco a los dos por todo el apoyo que me han dado para alcanzar el cumplimiento de mis metas.*

*A mi familia en general, porque siempre se han preocupado por mi aunque estemos lejos y por hacerme sentir que están ahí incondicionalmente.*

*Le agradezco a tres personas que aquí en la UCI conocí y son como otra familia para mí: Yaself, Susej y Rodnier. Personas increíbles que siempre estaré orgullosa de haber conocido, amigos incondicionales que en momentos difíciles, lejos de la familia de verdad, no me dejaron rendirme.*

*A mis tutores Andy, Coca, Susej, Ernesto y Esmaykel siempre supe que era muy afortunada de contar con la guía de ustedes.*

*A las personas que me ayudaron y me apoyaron en el trayecto, que no fueron pocas y aunque no las mencione tengo bien presente, este resultado no es solo mío, es de todos ustedes.*

*A mis compañeros de aula, fue un placer pasar estos años al lado de ustedes.*

*A todos los que de una forma u otra han incidido en mi vida.*

*Muchas gracias.*

## Declaración de autoría

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo. Para que así conste firmo la presente a los días del mes de del año .

\_\_\_\_\_  
Yaimara Fleites Reynaldo  
Autor

\_\_\_\_\_  
Andy Trujillo Rivero  
Tutor

\_\_\_\_\_  
Susej Beovides Luis  
Co-Tutor

\_\_\_\_\_  
Ernesto Leyva Piñeda  
Co-Tutor

\_\_\_\_\_  
Esmaykel Vázquez Ávila  
Co-Tutor

## RESUMEN

Estudios recientes indican el papel esencial que desempeña la actividad emocional si se desea que los computadores generen respuestas racionales ante una toma de decisiones. Este hecho combinado con la habilidad de los mismos en el reconocimiento de expresiones faciales, abre nuevas áreas para la investigación científica. El presente trabajo tiene como objetivo lograr clasificar automáticamente las emociones humanas, mediante el uso del dispositivo Kinect. Luego de un estudio del reconocimiento de las emociones básicas o universales solo se tienen en cuenta las emociones felicidad y tristeza, para probar la validez del modelo de clasificación desarrollado. Se analizaron algoritmos de Inteligencia Artificial con el uso de la herramienta WEKA para lograr la clasificación de dichas emociones, obteniendo los mejores resultados con el algoritmo de Optimización Secuencial Mínima (SMO). La aplicación propuesta brinda un modelo para clasificar emociones en tiempo real, según los movimientos faciales de las personas.

**Palabras claves:** Clasificación, Emociones, Expresiones faciales, Inteligencia Artificial, Kinect.

## Contenido

Resumen .....	VI
Índice de tablas.....	X
Índice de figuras .....	XII
Introducción .....	1
Capítulo 1. Fundamentación teórica .....	4
1.1    Introducción .....	4
1.2    Detección y caracterización de emociones .....	4
1.2.1    Características de las emociones .....	4
1.2.2    Detección de emociones.....	6
1.2.3    Notación de intensidad .....	8
1.3    Reconocimiento de patrones .....	9
1.4    Algoritmos de clasificación automática .....	10
1.4.1    K-Vecinos más cercanos .....	10
1.4.2    Árboles de decisión .....	11
Algoritmo C4.5 .....	11
1.4.3    Redes Neuronales .....	12
1.4.4    Máquinas de Soporte Vectorial.....	13
1.5    Herramientas y Metodologías .....	14
1.5.1    Kinect .....	14
Unidades de Animación .....	16
1.5.2    Biblioteca de integración.....	18
1.5.3    WEKA.....	19
1.5.4    Lenguaje de programación .....	19
1.5.5    Entorno de Desarrollo.....	19
1.5.6    Metodología de desarrollo .....	20
Conclusiones del capítulo.....	20
Capítulo 2. Propuesta de solución .....	21
2.1    Introducción .....	21
2.2    Descripción de la solución propuesta.....	21
2.3    Obtención de los datos de entrenamiento.....	23

2.3.1	Diseño de la captura de la información facial de la persona .....	23
2.3.2	Variante de diseño de la Obtención de datos de entrenamiento. ....	25
2.4	Selección del modelo de clasificación.....	26
2.5	Clasificación automática de emociones en las personas .....	28
2.6	Ingeniería de Software.....	28
2.6.1	Especificación de Requisitos .....	28
	Requisitos funcionales .....	29
	Requisitos no funcionales.....	29
2.6.2	Fase de exploración .....	30
	Descripción de requisitos .....	30
2.6.3	Fase de planificación .....	35
	Estimación de esfuerzo .....	36
	Plan de iteraciones.....	36
	Plan de entrega.....	37
2.6.4	Fase de diseño .....	37
	Arquitectura.....	38
	Patrones de diseño .....	40
	Tarjetas CRC .....	42
	Conclusiones del capítulo.....	44
Capítulo 3.	Análisis y Resultados.....	45
3.1	Introducción.....	45
3.2	Selección del Modelo de Clasificación .....	45
3.2.1	Resultados de la comparación entre ambos diseños .....	46
3.2.2	Mejoras implementadas.....	47
	Selección de atributos .....	47
	Filtros de pre-procesamiento.....	48
3.2.3	Algoritmo de Optimización Secuencial Mínima (SMO).....	49
3.3	Codificación .....	50
3.3.1	Estándares de codificación .....	50
	Nombres .....	50
	Indentación .....	51



---

Líneas y espacios en blanco .....	51
3.4 Archivos generados .....	52
3.4.1 Estructura de los ficheros .....	52
3.5 Pruebas .....	54
Conclusiones del Capítulo .....	59
Conclusiones .....	60
Recomendaciones .....	61
Referencias bibliográficas .....	62
Anexos.....	65

## ÍNDICE DE TABLAS

Tabla 1. Ejemplo de definiciones de las AU.....	7
Tabla 2 Definición de las emociones básicas según Paul Ekman (3).....	8
Tabla 3. Identificación de las Unidades de Animación y sus valores (21).....	17
Tabla 4. Correspondencia entre las emociones y los puntos reconocidos por el Kinect.....	24
Tabla 6: Requisitos Funcionales.....	29
Tabla 7. Historia de Usuario #1.....	30
Tabla 8. Historia de Usuario #2.....	31
Tabla 9. Historia de Usuario #3.....	31
Tabla 10. Historia de Usuario #4.....	32
Tabla 11. Historia de Usuario #5.....	32
Tabla 12. Historia de Usuario #6.....	32
Tabla 13. Historia de Usuario #7.....	33
Tabla 14. Historia de Usuario #8.....	33
Tabla 15. Historia de Usuario #9.....	34
Tabla 16. Historia de Usuario #10.....	34
Tabla 17. Historia de Usuario #11.....	35
Tabla 18. Estimación de esfuerzo por Historia de Usuario.....	36
Tabla 19. Plan de entregas.....	37
Tabla 20. Tarjeta CRC #1.....	43
Tabla 21. Tarjeta CRC #2.....	43
Tabla 22. Tarjeta CRC #3.....	43
Tabla 23. Tarjeta CRC #4.....	44
Tabla 25. Porcientos de instancias correctamente clasificadas con puntos faciales como datos.....	45
Tabla 26. Porcientos de instancias correctamente clasificadas con la variante de Datos.....	46
Tabla 24. Estructura del nombre de los archivos generados por el Módulo de Captura de Datos.....	52
Tabla 27. Caso de prueba de aceptación #1.....	54
Tabla 28. Caso de prueba de aceptación #2.....	54
Tabla 29. Caso de prueba de aceptación #3.....	55
Tabla 30. Caso de prueba de aceptación #4.....	55
Tabla 31. Caso de prueba de aceptación #5.....	56

---

Tabla 32. Caso de prueba de aceptación #6.....	56
Tabla 33. Caso de prueba de aceptación #7.....	57
Tabla 34. Caso de prueba de aceptación #8.....	57
Tabla 35. Caso de prueba de aceptación #9.....	58
Tabla 36. Caso de prueba de aceptación #10.....	58
Tabla 37. Caso de prueba de aceptación #11.....	59

## ÍNDICE DE FIGURAS

Figura 1. Componentes básicos de un sistema de PR (10).....	9
Figura 2.Arquitectura de una Red Neuronal Simple (14).....	13
Figura 3. Identificación de componentes del Kinect. ....	15
Figura 4. Rangos de efectividad del sensor en metros.....	15
Figura 5. Puntos en el rostro detectados por el Kinect según la página oficial de Microsoft (21).....	16
Figura 6. Fases de desarrollo del sistema para la clasificación de emociones. ....	22
Figura 7. Diagrama de flujo de procesos del sistema.....	22
Figura 8. Puntos re-enumerados como propuesta a utilizar. ....	24
Figura 9. Relación entre las AU de Ekman y las Unidades de Animación del Kinect, emoción: Felicidad. .	26
Figura 10.Relación entre las AU de Ekman y las Unidades de Animación del Kinect, emoción: Tristeza. ...	26
Figura 11. Arquitectura por capas. ....	39
Figura 12.Diseño de la Arquitectura de 3 capas del sistema.....	39
Figura 13. Ejemplo de utilización del patrón Experto.....	41
Figura 14. Ejemplo de utilización del patrón Creador. ....	41
Figura 15. Ejemplo de utilización del patrón Singleton. ....	42
Figura 16. Comportamiento de los algoritmos en WEKA según el juego de datos. ....	47
Figura 17. Resultado de la selección de atributos en WEKA.....	48
Figura 18. Matriz de Confusión para el algoritmo SMO.....	48
Figura 19. Comportamiento de los algoritmos en WEKA según los datos de las Unidades de Animación y sus modificaciones.....	49
Figura 20. Archivo de puntos faciales generado por la fase de Obtención de datos de entrenamiento. ....	53
Figura 21. Archivo de .ARFF generado por la fase de Obtención de datos de entrenamiento. ....	53

## INTRODUCCIÓN

El campo de la psicología ha alcanzado un grado de madurez elevado en cuanto al reconocimiento de expresiones faciales. Un gran número de prestigiosos psicólogos han realizado estudios en profundidad obteniendo información sólida sobre el reconocimiento de expresiones faciales y su vinculación con las emociones humanas.

El reconocimiento de las emociones se hace esencial en el aprendizaje de vías requeridas si se desea que las máquinas sean capaces de construir una respuesta racional. Para tener computadoras efectivas en la toma de decisiones, se necesitan mecanismos tales como la emoción que funcionen coordinadamente con sus sistemas. Este hecho y el avance de la tecnología acontecido en los últimos años han permitido llevar a cabo tareas que en el pasado hubieran sido impensables.

La Inteligencia Artificial como rama de la Informática que permite la realización automática de operaciones hasta ahora exclusivas de la inteligencia humana, ha sido objeto de estudio para lograr establecer esta analogía entre una actividad relativamente difícil para el ordenador y sencilla para el hombre.

En Cuba, el Centro de Entornos Interactivos 3D: Vertex de la Universidad de las Ciencias Informáticas desarrolló en conjunto con el Centro Nacional de Rehabilitación "Julio Díaz" ubicado en La Habana, el sistema informático "Danzoterapia v1.0". Esta herramienta consiste en un tutor virtual que realiza demostraciones animadas de los ejercicios que deben ejecutar los pacientes con limitaciones físico-motoras, como parte de su terapia de rehabilitación.

Aunque los resultados preliminares son positivos, la primera versión de Danzoterapia no identifica los estados de ánimo de los usuarios. En ocasiones los ejercicios se pueden tornar repetitivos y monótonos, lo que pudiera provocar rechazo, abandono o mala ejecución de la terapia por parte de los pacientes. El sistema adolece de retroalimentación efectiva en este sentido y no es capaz de estimular a los pacientes cuando realizan progresos.

Por todo lo planteado y teniendo en cuenta que el reconocimiento de emociones se puede aplicar en campos tan diversos como la psicología, la enseñanza o el marketing, surge la motivación de dar solución al siguiente **problema de investigación**:

¿Cómo identificar automáticamente emociones en las personas, basado en sus expresiones faciales?

Para dar solución al problema se traza como **objetivo general** de la investigación: Clasificar automáticamente emociones en las personas, a partir de información adquirida con el dispositivo Kinect.

Definiéndose como **objeto de estudio** el reconocimiento de patrones y como **campo de acción** la clasificación automática de emociones.

Se definieron las siguientes tareas de investigación para alcanzar el resultado deseado:

1. Elaboración del marco teórico de la investigación a partir del estado del arte existente.
  - 1.1. Caracterización de las técnicas de reconocimiento facial y de clasificación automática de estados de ánimo, descritas en la literatura, para determinar aspectos relevantes que puedan ser útiles en la solución al problema de esta investigación.
  - 1.2. Caracterización del problema de clasificación automática de estados de ánimo, a partir de datos del rostro humano capturados con el Kinect, para determinar qué elementos intervienen en él y cómo modelarlo.
2. Diseño, implementación y prueba del proceso de captura de datos del problema.
  - 2.1. Implementación y prueba de algoritmos de selección, transformación y limpieza de datos para simplificar el proceso de clasificación y garantizar la calidad de los datos al eliminar errores, inconsistencias, redundancias y datos incompletos.
3. Diseño, implementación y prueba del proceso de clasificación automática de emociones en las personas, basado en sus expresiones faciales.
  - 3.1. Experimentación con algoritmos de clasificación automática para obtener modelos computacionales que permitan identificar emociones de una persona, a partir de sus características faciales.
  - 3.2. Experimentación con el método de validación cruzada; para comprobar la capacidad de generalización de los modelos computacionales de clasificación ante nuevas instancias.
  - 3.3. Implementación y prueba de las funcionalidades que permiten cargar y utilizar el modelo de clasificación automática obtenido, para identificar emociones de una persona a partir de sus características faciales.

Como parte del cumplimiento de las tareas definidas se emplearon los siguientes **métodos científicos**:

1. Teóricos:

- Histórico - Lógico: este método se utilizó para el análisis del estado del arte, obteniéndose los aspectos más significativos sobre los estudios de las emociones humanas, las distintas técnicas de reconocimiento facial y los algoritmos de clasificación de Inteligencia Artificial relevantes.
- Modelación: mediante este método se modeló el prototipo funcional de la aplicación y su arquitectura.

2. Empírico:

- Estudio de la documentación: este método se utilizó para realizar la búsqueda exhaustiva de la información referente a los distintos temas tratados en el estudio del estado del arte.
- Test o prueba: se utilizó para evaluar la calidad de los datos obtenidos en el proceso de captura de datos y en la validación de los resultados alcanzados por los algoritmos de clasificación. Además, para comprobar que las funcionalidades del sistema tengan las salidas esperadas.
- Experimentación: se experimentó con varios algoritmos de clasificación y juegos de datos para determinar un modelo de clasificación de emociones adecuado.

El presente documento está estructurado en 3 capítulos. En el Capítulo 1 se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. En el Capítulo 2 se propone un método de solución basado en la captura de las expresiones faciales. En el Capítulo 3 se muestran los resultados de la propuesta desarrollada y se exponen las pruebas que se realizaron.

# CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

## 1.1 INTRODUCCIÓN

El presente capítulo abarca el estudio realizado sobre el proceso de clasificación de emociones en los humanos. Se realiza un análisis de las emociones y sus formas de detección para la extracción de rasgos faciales relevantes. Además, se caracterizan los procesos de captura de la información facial para su posterior tratamiento, el reconocimiento de patrones y los algoritmos de Inteligencia Artificial que resuelven problemas de clasificación.

## 1.2 DETECCIÓN Y CARACTERIZACIÓN DE EMOCIONES

El estudio sobre el reconocimiento de expresiones faciales y emociones humanas por parte de numerosos psicólogos ha obtenido grandes resultados, validados por experimentos en situaciones reales. Una emoción es definida por la Real Academia de la Lengua Española (RAE) como: “la alteración de ánimo intensa y pasajera, agradable o penosa que va acompañada de cierta conmoción somática” (1). Las emociones, según el psicólogo Paul Ekman pionero en el reconocimiento de expresiones y micro expresiones faciales, son proyectadas independientemente del entorno sociocultural del individuo y de sus características fisiológicas encontrándose de esta forma un patrón general en las emociones más básicas (2) (3).

### 1.2.1 CARACTERÍSTICAS DE LAS EMOCIONES

Todas las emociones tienen alguna función que les confiere utilidad y permite que el hombre ejecute con eficacia las reacciones conductuales apropiadas. Incluso las emociones más desagradables tienen funciones importantes en la adaptación social (4). El conocimiento de algunas de las características de estas tales como motivos que las provoca, consecuencias y rasgos detectados durante su aparición puede ser útiles para su reconocimiento.

A continuación se enuncian algunas de las características principales de las 6 emociones básicas o universales.



---

**Felicidad:**

- Estado placentero, deseable, sensación de bienestar, autoestima y autoconfianza.
- Favorece los vínculos sociales y relaciones interpersonales.
- Favorece la recepción e interpretación positiva de los diversos estímulos ambientales.
- Causada por el logro, consecución exitosa de los objetivos que se pretenden.
- Favorece el rendimiento cognitivo, solución de problemas y creatividad, así como el aprendizaje y la memoria.

**Ira:**

- Estimulación aversiva, tanto física o sensorial, como cognitiva<sup>1</sup>.
- Condiciones que generan frustración, interrupción de una conducta motivada, situaciones injustas, o atentados contra valores morales.
- Incapacidad o dificultad para la ejecución eficaz de procesos cognitivos.
- Movilización de energía para las reacciones de autodefensa o de ataque.
- Sensación de energía e impulsividad, necesidad de actuar de forma intensa e inmediata (física o verbalmente) para solucionar de forma activa la situación problemática.

**Miedo:**

- Es una de las reacciones que produce mayor cantidad de trastornos mentales, conductuales y emocionales.
- Procesos de valoración secundaria que interpretan una situación como peligrosa.
- Al prestar una atención casi exclusiva al estímulo temido, facilita que el organismo reaccione rápidamente ante el mismo.
- Se trata de una de las emociones más intensas y desagradables. Genera aprensión, desasosiego y malestar.

**Tristeza:**

- Separación física o psicológica, pérdida o fracaso, decepción, especialmente si se han desvanecido esperanzas puestas en algo.
- Focalización de la atención en las consecuencias a nivel interno de la situación.

---

<sup>1</sup> Del conocimiento o relacionado con él.

- Provoca desánimo, melancolía, desaliento, pérdida de energía.
- Puede llegar a generar estados depresivos agudos y baja autoestima.

### **Sorpresa:**

- Presencia de estímulos novedosos débiles o moderadamente intensos, acontecimientos inesperados.
- Interrupción de la actividad que se está realizando en ese momento.
- Atención y memoria de trabajo dedicadas a procesar la información novedosa.
- Facilita los procesos atencionales, conductas de exploración e interés por la situación novedosa.

### **Desagrado:**









- Estímulos desagradables (químicos fundamentalmente) potencialmente peligrosos o molestos.
- Generación de respuestas de escape o evitación de situaciones desagradables.
- Produce necesidad de evitación o alejamiento del estímulo. Si el estímulo es oloroso o gustativo aparecen sensaciones gastrointestinales desagradables, tales como náusea (4).

Conocer las reacciones tanto físicas como internas de cada emoción constituye un paso importante para su detección. Un punto de partida, son las reacciones orgánicas. Son muchos los parámetros físicos que pueden hacer notar la presencia de una emoción, tales como: expresiones faciales, distribución de la sangre, ritmo del corazón, respiración, secreción salival, entre otras.

## **1.2.2 DETECCIÓN DE EMOCIONES**

En el año 2002, Paul Ekman realizó un aporte importante a los estudios de reconocimiento de expresiones faciales humanas, obteniendo el Sistema de Codificación de la Acción Facial (FACS, por sus siglas en inglés) basado en un sistema desarrollado originalmente por el anatomista sueco Carl-Herman Hjortsjö. Este sistema codifica los movimientos de los músculos faciales indistintamente de los diferentes cambios momentáneos en la apariencia facial reflejándolos en las denominadas Unidades de Acción (AU, según sus siglas en inglés) (3) (5). La Tabla 1 muestra algunas de las AU y el movimiento facial al que hacen referencia. Una lista completa de las AU se puede encontrar en el artículo "Sistema de Codificación de Acción facial - Lista de Unidades de Acción" (5).

Tabla 1. Ejemplo de definiciones de las AU.

AU	Descripción	Imágenes
1	Levantamiento interior de cejas	
2	Levantamiento exterior de cejas	
4	Bajar cejas	
6	Levantamiento de mejillas	
9	Arrugar la nariz	
12	Tiramiento labial esquinal	
13	Tiramiento labial frontal	
15	Depresión labial esquinal	

Mediante el uso de las AU, fue posible codificar expresiones ambiguas y sutiles, por lo tanto FACS es potencialmente más adecuado para el análisis de las pequeñas diferencias en rasgos faciales. Sin embargo, era un sistema diseñado para realizar un análisis de *frame-by-frame* y por lo tanto inútil para las pruebas en tiempo real. En el documento “*Automated Facial Expression Recognition System*” (6), se explica cómo “un entrenado, certificado codificador FACS tarda una media de 2 horas en codificar 2 minutos de video”. Por esta razón, Friesen y Ekman también desarrollaron EMFACS (Emoción FACS) que utiliza los mismos principios de FACS con la diferencia de centrarse en el reconocimiento de ciertas emociones. De esta manera sólo se utilizan pocas AU. Luego, el cálculo realizado se reduce drásticamente lo que permite analizar las expresiones faciales en tiempo real (7). Si se desea reconocer gran cantidad de emociones es

aconsejable tener en cuenta FACS, pero de no ser así, puede ser productivo guiarse por la idea de EMFACS, de limitarse a seleccionar las AU que aporten valor real.

### 1.2.3 NOTACIÓN DE INTENSIDAD

Los movimientos de la cara pueden ser menos o más exagerados, para esto Ekman define una notación de intensidad que se acompaña de la AU correspondiente de ser necesario. Las intensidades de FACS se anotan añadiendo las letras A-E (para las intensidades mínima-máxima respectivamente) y números para la AU (por ejemplo, AU 1A es la traza más débil de AU 1, por su parte AU 1E es la máxima intensidad posible para el gesto facial) (8).

- A Rastro
- B Leve
- C Pronunciado
- D Severo o extremo
- E Máximo

La Tabla 2 muestra ejemplos de la codificación de las emociones según Ekman donde la presencia de un conjunto de AU equivale a una emoción.

Tabla 2 Definición de las emociones básicas según Paul Ekman (3).

Emoción	Unidad de Acción
Disgusto	9+15+16
Felicidad	6+12
Ira	4+5+7+23
Tristeza	1+4+15
Sorpresa	1+2+5B+26
Miedo	1+2+4+5+7+20+26

Para que las emociones lleguen a ser datos que un sistema pueda utilizar, es necesario además de identificar los elementos distintivos que las caracterizan, realizar un estudio de las técnicas informáticas que permiten el reconocimiento de la información facial y su tratamiento.

## 1.3 RECONOCIMIENTO DE PATRONES

El reconocimiento de patrones (PR, por sus siglas en inglés) es la disciplina científica cuyo objetivo es la clasificación de los objetos en una serie de categorías o clases. Dependiendo de la aplicación, estos objetos pueden ser imágenes o formas de onda de señal o cualquier tipo de mediciones que deben ser clasificados (9).

Para el PR se siguen los siguientes procesos:

1. Adquisición de datos.
2. Extracción de características.
3. Toma de decisiones.

Un sistema de PR incluye:

- Un sensor que recoge fielmente los elementos del universo a ser clasificados.
- Un mecanismo de extracción de características cuyo propósito es extraer información útil, eliminando la información redundante e irrelevante.
- Una etapa de toma de decisiones en la cual se asignan los datos a la categoría apropiada de los patrones de la clase desconocida *a priori* (10).

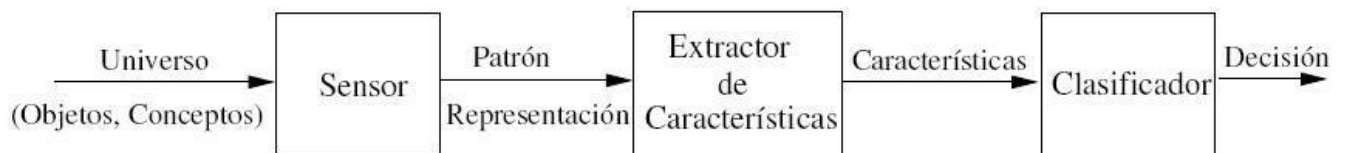


Figura 1. Componentes básicos de un sistema de PR (10).

**Sensor:** es el dispositivo encargado de la adquisición de datos. Ha de ser capaz de transformar magnitudes físicas o químicas, llamadas variables de instrumentación, en magnitudes eléctricas, que pueden ser distancia, aceleración, inclinación, desplazamiento, entre otras.

**Extracción de características:** genera características que puedan ser usadas en el proceso de clasificación de los datos. En ocasiones viene precedido por un pre-procesado de la señal, necesario para corregir posibles deficiencias en los datos debido a errores del sensor, o bien para preparar los datos de cara a procesos posteriores. Consiste en seleccionar cuál es el tipo de característica o rasgo más distintivo

para describir los objetos. Para ello, se deben localizar los rasgos que inciden en el problema de manera determinante. Esta etapa también puede ser diseñada dentro de la clasificación.

**Clasificador:** encargado de la clasificación de rasgos faciales, trata de asignar las diferentes partes del vector de las características<sup>2</sup> a grupos o clases, basándose en las características extraídas. En esta etapa se usa lo que se conoce como aprendizaje automático, cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender (10).

Según se tenga constancia o no de un conjunto previo de datos que permita al sistema aprender, la clasificación puede ser supervisada o no supervisada.

**Clasificación no supervisada:** también conocida como clasificación sin aprendizaje. Las clases de los objetos no son conocidas *a priori*. Se recurre a un agrupamiento natural (“*clustering*”).

**Clasificación supervisada:** también es conocida como clasificación con aprendizaje. Se tiene una muestra de la que se conoce *a priori* la clase a la que pertenecen y que servirán para generar una medida cuantitativa de las características de cada una de las clases (11).

## 1.4 ALGORITMOS DE CLASIFICACIÓN AUTOMÁTICA

Algunos de los algoritmos más utilizados que resuelven problemas de clasificación se describen a continuación.

### 1.4.1 K-VECINOS MÁS CERCANOS

El algoritmo K-Vecinos más cercanos (K-NN, por sus siglas en inglés) es una técnica supervisada no paramétrica, puesto que no se conoce la distribución que presentan los patrones. Por lo tanto se estima el valor de la función densidad o la probabilidad *a posteriori* de que un patrón  $X$  pertenezca a la clase  $W$  directamente a partir de la información proporcionada por el conjunto de patrones (12).

K-NN consta de tomar los  $k$  patrones más cercanos al que se quiere clasificar (patrón  $X$ ), delimitándose así un círculo con centro  $X$ . Hay tres elementos clave de este enfoque: un conjunto de objetos clasificados o

---

<sup>2</sup> Conjunto de propiedades que distinguen a los objetos.

etiquetados, una distancia o similitud métrica para calcular la distancia entre los objetos y el valor de  $k$  (número de vecinos más próximos). Para clasificar el patrón  $X$ , se calcula la distancia de este a los  $k$  patrones etiquetados, se identifican sus  $k$ -vecinos más cercanos y las etiquetas de clase de estos se utilizan para determinar la etiqueta de clase del patrón  $X$  (13).

La idea principal que se maneja con esta técnica de estimación es que el área del círculo que encierra los  $k$  patrones más cercanos será menor en regiones densamente pobladas y mayor en donde los patrones están más dispersos.

Como desventajas se puede observar que la misma hace todo el trabajo en tiempo de ejecución, por lo que el algoritmo puede ser muy lento en el caso de conjuntos muy grandes. Además, es muy sensible a características redundantes porque estas colaboran para la proximidad entre los patrones (12).

#### 1.4.2 ÁRBOLES DE DECISIÓN

Los árboles de decisión son herramientas de clasificación sin métrica que permiten analizar decisiones secuenciales. Se basan en el uso de resultados y en probabilidades asociadas y usualmente se les representa con notación de grafos para poder visualizarlos. La idea principal es que dado un conjunto de datos etiquetados  $(X_n y_n)$  con  $1 \leq n \leq N$ , donde  $X_n \in R^n$  y  $Y_n \in C$ ,  $c=1 \dots C$  lo que se quiere es construir un modelo de clasificación en el cual a cada nuevo dato  $X_n$  podamos asociarle una clase  $C$ .

Se puede optar por la construcción de árboles de decisión binarios, donde cada nodo tiene exactamente 2 hijos o ninguno (las hojas), o árboles  $n$ -arios donde los nodos tienen  $n$  cantidad de hijos. Los primeros tienen la ventaja de que su construcción es más fácil de automatizar utilizando algoritmos computacionales, de todas formas, es posible representar cualquier tipo de árbol como uno binario.

Los árboles de decisión son algoritmos inestables, pequeñas modificaciones en el conjunto de entrenamiento pueden generar grandes cambios en el clasificador. Por otra parte, este tipo de clasificadores son intuitivos ya que el proceso de decisión puede ser generado a partir de decisiones simples. Además, para construir árboles de decisión se pueden utilizar tanto características cualitativas como cuantitativas y tienen la ventaja de que se pueden aplicar tanto a problemas discretos como continuos (12).

#### ALGORITMO C4.5

En este algoritmo para cada nodo se elige un atributo de los datos que divida el conjunto de muestras de forma más eficiente. Es decir, que particione los datos en buenos subconjuntos de una clase u otra. El

criterio que utiliza es el de ganancia de información o diferencia de entropía, que se utiliza para elegir el atributo que dividirá los datos. El atributo que posea la mayor ganancia de información normalizada se elige como parámetro de decisión. El algoritmo continúa de forma recursiva subdividiendo cada conjunto generado en el paso anterior. Para este algoritmo hay tres pasos base para la recursión, éstos son:

- Todas las muestras en la lista pertenecen a la misma clase. Cuando esto sucede, se crea un nodo de hoja para el árbol de decisión diciendo que elija esa clase.
- Ninguna de las características proporciona ninguna ganancia de información. En este caso, se crea un nodo de decisión más arriba del árbol utilizando el valor esperado de la clase.
- Detección de una instancia de la clase previamente no vista. Una vez más, se crea un nodo de decisión más arriba en el árbol con el valor esperado (12) (13).

### 1.4.3 REDES NEURONALES

---

Las Redes Neuronales Artificiales (ANN, por sus siglas en inglés) están inspiradas en las redes neuronales biológicas del cerebro humano. Constituidas por elementos que se comportan de forma similar a la neurona biológica en sus funciones más comunes. Estos elementos están organizados de una forma parecida a la que presenta el cerebro humano.

En las ANN, la unidad análoga a la neurona biológica es el elemento procesador (PE, por sus siglas en inglés). Un PE tiene varias entradas y las combina, normalmente con una suma básica. La suma de las entradas es modificada por una función de transferencia y el valor de la salida de esta función de transferencia se pasa directamente a la salida del elemento procesador. La salida del PE se puede conectar a las entradas de otras neuronas artificiales (PE) mediante conexiones ponderadas correspondientes a la eficacia de la sinapsis de las conexiones neuronales.

Generalmente los PE están organizados en grupos llamados niveles o capas. Una red típica consiste en una secuencia de capas con conexiones entre capas adyacentes consecutivas. Existen dos capas con conexiones con el mundo exterior. Una capa de entrada, buffer de entrada, donde se presentan los datos a la red y una capa buffer de salida que mantiene la respuesta de la red a una entrada. El resto de las capas reciben el nombre de capas ocultas. En la Figura 2, se muestra el aspecto de una ANN Simple (14).



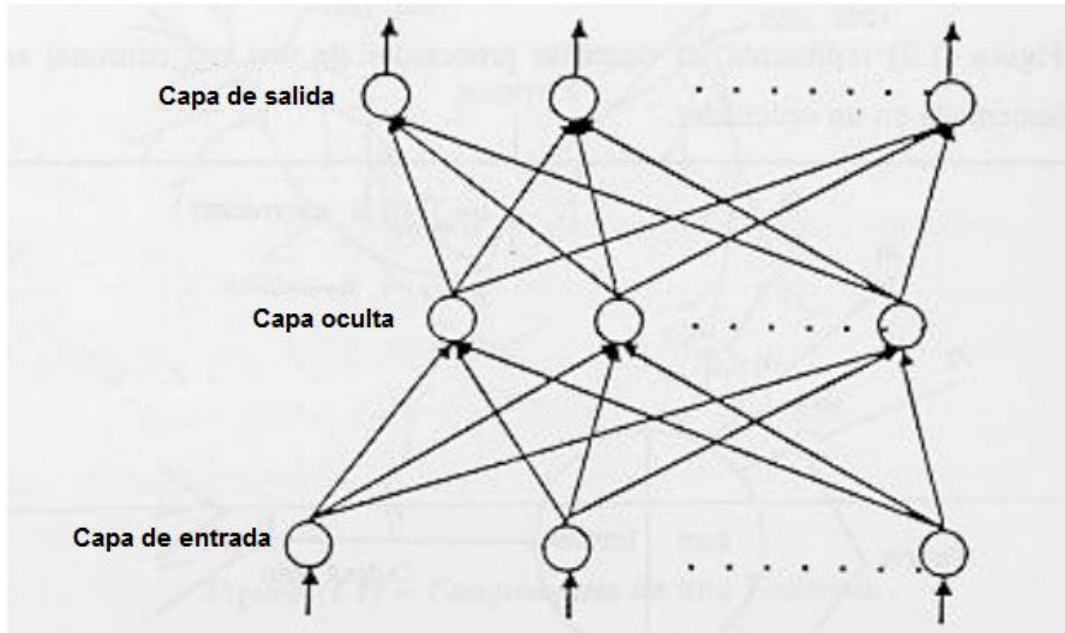


Figura 2.Arquitectura de una Red Neuronal Simple (14).

Las ANN al margen de "parecerse" al cerebro presentan una serie de características propias del cerebro. Por ejemplo, las ANN aprenden de la experiencia, generalizan de ejemplos previos a ejemplos nuevos y abstraen las características principales de una serie de datos.

Aprenden debido a que estas pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir salidas consistentes. Las ANN generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión. Algunas ANN son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos (14).

#### 1.4.4 MÁQUINAS DE SOPORTE VECTORIAL

Las Maquinas de Soporte Vectorial (SVM, por sus siglas en inglés) son un tipo de clasificadores de patrones basados en técnicas estadísticas de aprendizaje. El método de las SVM permite abordar de manera general la resolución de problemas de clasificación y de regresión.

La idea consiste en transformar el conjunto de vectores de entrada  $X = \{x_i | x_i = (x_{i1}, x_{i2}, \dots, x_{in})\}$  (patrones n-dimensionales) en otro conjunto de vectores  $Y$  de una dimensión más alta (incluso de dimensión infinita) en los que el problema pueda solucionarse linealmente. Cuando las clases no son linealmente separables transforma los vectores de entrada en vectores de mayor dimensión para conseguir que las clases sean linealmente separables.

Este clasificador presenta una gran velocidad de entrenamiento, permitiendo construir hiperplanos óptimos de separación en problemas de clasificación. Aproxima la probabilidad de que un patrón de entrada pertenezca o no a una clase, mediante funciones (relaciones) entre los datos de entrada. Para ello utiliza el método basado en funciones de *kernel*, la similitud entre los datos se expresa a través de la noción de *kernel*, definido según el problema a resolver.

Las SVM están a la cabeza de los métodos de clasificación por permitir construir fronteras de decisión flexibles y por su buena capacidad de generalización (13).

## 1.5 HERRAMIENTAS Y METODOLOGÍAS

### 1.5.1 KINECT

Para la captura y recopilación de datos se hará uso del Kinect, el cual es un dispositivo creado por Microsoft Corporation, con el objetivo de permitir la interacción en tiempo real de los usuarios con la videoconsola Xbox 360. Dentro de sus características se puede destacar la capacidad de escaneo 3D, lo que permite reconstruir una escena a todo color. El Kinect es un hardware más barato que los sistemas complejos habituales con varias cámaras y está pensado como un sistema de captura de movimiento sin marcadores 3D ya que puede mostrar un esqueleto simplificado sin hacer uso de vestimenta especial u otro dispositivo (15).

El impacto de Kinect se ha extendido mucho más allá de la industria de videojuegos. Con su amplia disponibilidad y bajo costo, muchos investigadores y profesionales de informática, ingeniería electrónica y robótica están aprovechando la tecnología de sensores para desarrollar nuevas formas creativas de interactuar con máquinas y para realizar otras tareas.

El sensor Kinect incorpora hardware de detección avanzada. En particular, contiene un sensor de profundidad, una cámara de color y una matriz de cuatro micrófonos que proporcionan todo el cuerpo captura de movimiento 3D, reconocimiento facial y capacidades de reconocimiento de voz (16).

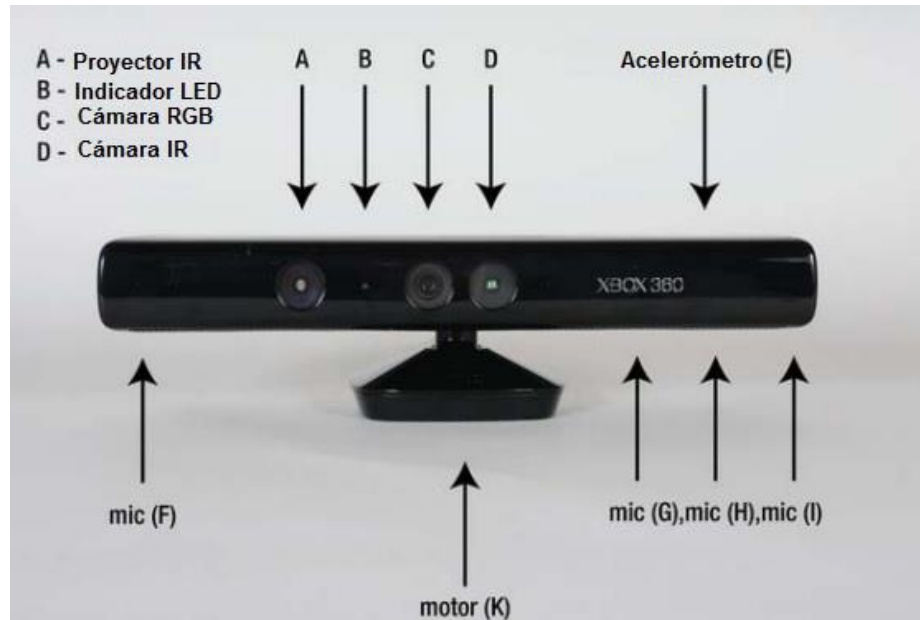


Figura 3. Identificación de componentes del Kinect.

Salida: A) IR (infrarrojo) proyector con láser de estructura ligera, B) Indicador LED y K) motor para controlar la inclinación de la base. Entrada: F-I) cuatro micrófonos, C-D) dos cámaras (RGB e IR) y E) un acelerómetro (17).

El sensor de profundidad cuenta con 2 modos, por defecto y cercano, con un rango válido que va desde 0.8 metros hasta 4 metros de distancia y desde los 0.4 metros hasta 3 metros de distancia correspondientemente, tal como se muestra en la Figura 4 (18).

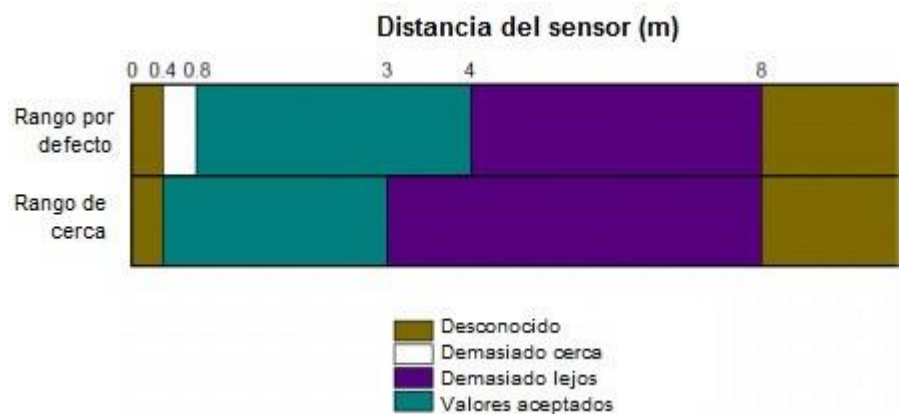


Figura 4. Rangos de efectividad del sensor en metros.

El *Software Development Kit* (SDK) contiene una serie de bibliotecas que permiten obtener información facial en tiempo real a partir de los datos que es capaz de capturar la cámara.

El motor de seguimiento facial de Kinect analiza los datos capturados por la cámara, deduce la posición de la cara y las principales características del rostro y hace esa información disponible para cualquier aplicación en tiempo real.

Según Microsoft Corporation, Kinect es capaz de detectar 87 puntos del rostro como se muestran en la Figura 5, estos puntos se devuelven en una matriz y se definen en el espacio de coordenadas de la imagen RGB (de 640 x 480 píxeles de resolución) devueltos desde el sensor.

Además, reconoce otros 13 puntos adicionales que no se muestran en la figura, incluye el centro del ojo, las esquinas de la boca y el centro de la nariz (19).

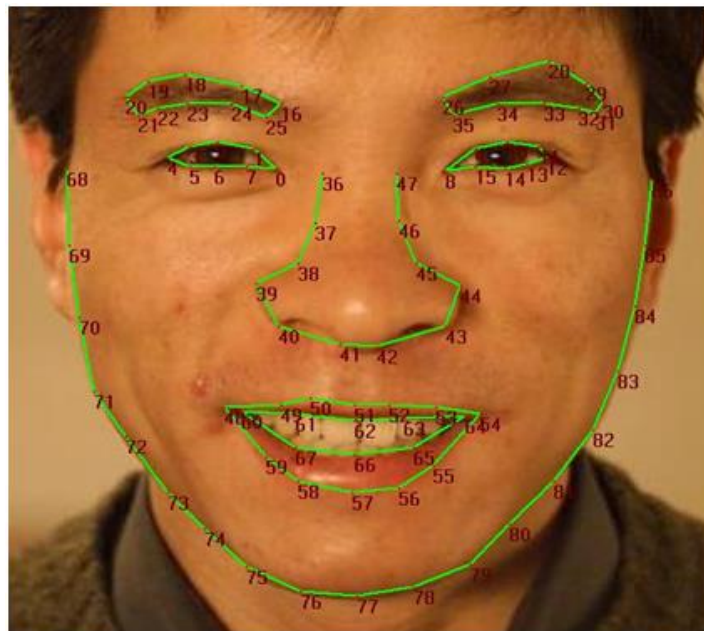


Figura 5. Puntos en el rostro detectados por el Kinect según la página oficial de Microsoft (19).

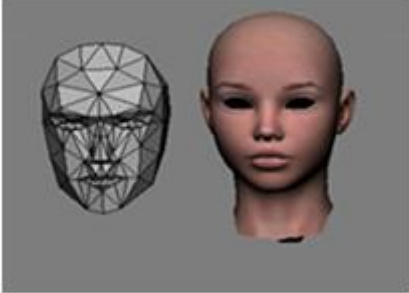



---



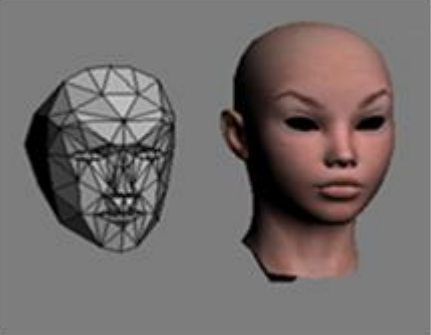
## UNIDADES DE ANIMACIÓN

---

La clase *FaceTracking* de la SDK del Kinect, encargada de capturar estos resultados, también es capaz de obtener los valores de 6 unidades de animación, que son un subconjunto de lo que se define en el modelo Candide3 (20). Estas Unidades de Animación brindan información sobre la apariencia facial en tiempo real. Cada Unidad de Animación se expresa como un peso numérico que varía entre -1 y 1. Véase la Tabla 3.

Tabla 3. Identificación de las Unidades de Animación y sus valores (19).

Nombre e identificador	Avatar	Valores
Rostro Neutro		<p>Todas las unidades de animación en 0.</p>
AU1- Elevación del labio superior.		<p>0 = neutro, que cubren los dientes.            1 = muestra los dientes totalmente.            -1 = labio empujado hacia abajo lo máximo posible.</p>
AU2- Mandíbula inferior.		<p>0 = cerrada.            1 = completamente abierta.            -1 = cerrada, al igual que 0.</p>
AU3- Estiramiento de labios.		<p>0 = neutro.            1 = totalmente estirada (La sonrisa de Joker).            -0.5 = redondeado (puchero).            -1 = totalmente redondeada (forma de beso).</p>

<p>AU4 - Cejas Bajas.</p>		<p>0 = neutro.  -1 = Levantado completamente.  +1 = completamente bajada (hasta el límite de los ojos).</p>
<p>AU5 – Depresión de la esquina de los labios.</p>		<p>0 = neutro.  -1 = sonrisa (muy feliz).  +1 = muy triste (ceño fruncido).</p>
<p>AU6 – Cejas exteriores elevadas.</p>		<p>0 = neutro  -1 = completamente bajada como una cara muy triste.  +1 = planteadas como en una expresión de profunda sorpresa.</p>

### 1.5.2 BIBLIOTECA DE INTEGRACIÓN

La SDK para la interacción con el Kinect es la biblioteca “*Kinect for Windows*” en su versión 1.8. Esta ofrece un conjunto de herramientas y funcionalidades para el desarrollo de aplicaciones capaces de utilizar los datos obtenidos por el Kinect de una forma rápida y sencilla. Además, permite crear aplicaciones que pueden rastrear los rostros humanos en tiempo real. El motor de seguimiento de la cara del SDK analiza los

---

datos de la cámara Kinect, deduce la representación de la cabeza y las expresiones faciales, y hace que la información esté disponible para una aplicación en tiempo real (21).

### 1.5.3 WEKA

---

WEKA (*Waikato Environment for Knowledge Analysis*) es una herramienta de código abierto para la minería de datos, desarrollada con tecnología Java. Tiene un conjunto de algoritmos de aprendizaje automático para tareas de minería de datos. Estos se pueden aplicar directamente a un conjunto de datos o desde una aplicación Java usando WEKA como biblioteca. Contiene herramientas para el procesamiento previo de datos, clasificación, regresión, *clustering*, reglas de asociación y su visualización. También es muy adecuado para el desarrollo de nuevos sistemas de aprendizaje de máquina. Funciona en los sistemas operativos Windows, GNU/Linux y Mac. Por sus características es usado para investigaciones, en la formación y el desarrollo de aplicaciones (22).

### 1.5.4 LENGUAJE DE PROGRAMACIÓN

---

El lenguaje que se utilizará es C# (“*C Sharp*” en inglés), el cual es un lenguaje de propósito general diseñado por Microsoft Corporation para su plataforma *.NET*. Es un lenguaje de programación orientado a objetos, con seguridad de tipos, control de versiones, eventos, recolección de elementos no utilizados y admite herencia única. El lenguaje de programación C#, además de combinar las mejores características de lenguajes preexistentes como Visual Basic, Java y C++, es el lenguaje en que está implementado el SDK del Kinect (23).

### 1.5.5 ENTORNO DE DESARROLLO

---

El Entorno de Desarrollo Integrado (IDE, por sus siglas en inglés) a emplear es Microsoft Visual Studio en su versión 12.0.21005.1 del año 2013. Este integra un amplio conjunto de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y móviles. Además, soporta múltiples lenguajes de programación tales como Visual Basic, C# y C++, los cuales son capaces de utilizar las potencialidades del Framework *.NET*. Además, es capaz de usar la SDK “*Kinect for Windows*”, logrando que la interacción con el Kinect sea fácil y sencilla (21).

### 1.5.6 METODOLOGÍA DE DESARROLLO

Para el desarrollo de la aplicación, debido al hecho de que está orientada a la producción de código, con ciclos muy cortos de desarrollo, que cuenta con un equipo pequeño de trabajo en un ambiente que puede presentar cambios durante su realización e involucra activamente al cliente en el proceso de desarrollo se decidió utilizar Programación Extrema (XP, por sus siglas en inglés), una metodología de procesos ágiles para el desarrollo de software. Es de las metodologías más exitosas debido a que se centra en potenciar las relaciones entre los desarrolladores para lograr el éxito, promoviendo el trabajo en equipo y la comunicación con el cliente, lo cual propicia un buen entorno de trabajo (24). Algunas de las características que la distinguen son:

- Desarrollo iterativo e incremental.
- Integración del equipo de programación con el usuario.
- Propiedad del código compartida.
- Corrección de errores.

## CONCLUSIONES DEL CAPÍTULO

- Se seguirá la teoría propuesta por el psicólogo Paul Ekman para el reconocimiento de las emociones a través de los rasgos faciales.
- De las 6 emociones básicas, solo se tomarán en cuenta las emociones de alegría y tristeza, para probar la validez del modelo de clasificación.
- Se utilizará para la captura de la información facial el dispositivo Kinect con la biblioteca *Kinect for Windows*.
- El tratamiento de datos y la obtención del modelo de clasificación se realizará con el apoyo de la herramienta WEKA.
- Se utilizará como lenguaje de programación C# y como IDE el Visual Studio 2013, para la automatización de los procesos de clasificación de emociones.
- Como metodología para el desarrollo se adoptará la Programación Extrema.



## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

### 2.1 INTRODUCCIÓN

En el presente capítulo se brindan detalles de la propuesta de solución desde el proceso de captura de datos y su preparación como conjunto de entrenamiento, hasta la selección del modelo de clasificación y su posterior implementación. Se identifican los elementos de la Ingeniería de Software a tener en cuenta para la construcción de la solución como el diseño del sistema y la planificación del proyecto.

### 2.2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

El problema a resolver está definido dentro del área de PR como un problema de clasificación. Se puede ver como un esclarecimiento de una dependencia, en la que el atributo dependiente puede tomar un valor entre varias clases ya conocidas. Para este caso las variables independientes son los datos del rostro y las clases, las emociones de felicidad y tristeza.

La propuesta de solución se divide en tres sub-procesos o fases, análogos a los definidos para el PR:

1. Obtención de los datos de entrenamiento.
2. Generación del modelo computacional de clasificación automática.
3. Clasificación automática de emociones en las personas

En la primera fase, se realiza la obtención de la información facial de la persona, se seleccionan los rasgos distintivos y son clasificados manualmente. Los datos generados sirven como base para la selección del modelo computacional de clasificación automática en la segunda fase, con el apoyo de la herramienta WEKA. Se encarga de actividades como el pre-procesamiento de los datos y la evaluación del comportamiento de los algoritmos implementados en WEKA con estos. Este modelo se utiliza para lograr la clasificación automática de los estados de ánimos en tiempo real en la tercera fase. La Figura 6 muestra cómo sería el proceso general para la clasificación automática de emociones.

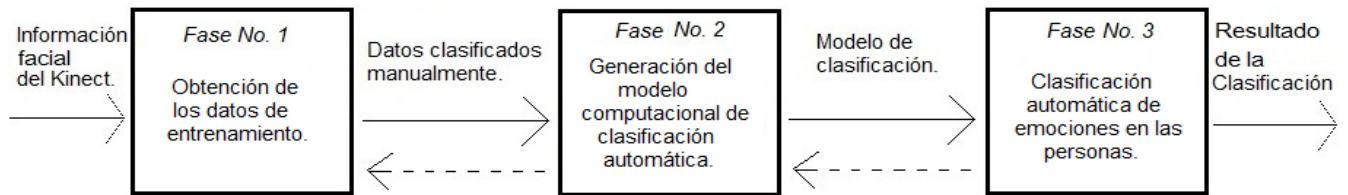


Figura 6. Fases de desarrollo del sistema para la clasificación de emociones.

El sistema podrá utilizarse en dos modos. Un modo destinado a la clasificación manual de los datos y creación del modelo de clasificación (entrenamiento), para el cual es necesario seguir los pasos, hasta la segunda fase, anteriormente descritos. Otro modo para la clasificación en tiempo real de las emociones con el modelo generado (ver diagrama de flujo de procesos Figura 7).

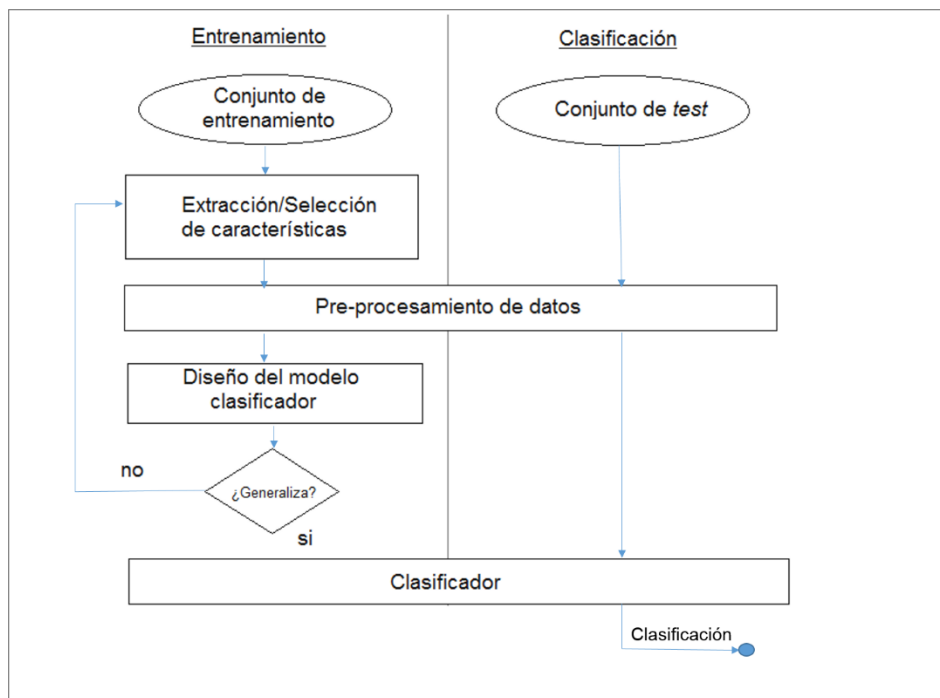


Figura 7. Diagrama de flujo de procesos del sistema.

Para el correcto funcionamiento del sistema descrito es necesario tener en cuenta que está diseñado para clasificar las emociones de una única persona en cada momento. Esta debe estar a una distancia aproximada entre 1 y 3 metros; estando el rostro de la persona relativamente de frente a la cámara del dispositivo Kinect.

El sistema propuesto es factible solo para proyectos que cumplan con las características anteriores, bajo otras condiciones requiere modificaciones para poder aplicarse.

## 2.3 OBTENCIÓN DE LOS DATOS DE ENTRENAMIENTO

Para poder realizar el reconocimiento de emociones es necesario implementar una fase de obtención de los datos que describan al patrón que se desea clasificar. Como mecanismo para la adquisición de la información facial y las imágenes se utiliza el dispositivo Kinect, este facilita en gran medida la adquisición de los datos necesarios para realizar el análisis de las expresiones faciales.

Con este objetivo se diseña la fase de Obtención de los datos de entrenamiento, la cual le permite al usuario tomar muestras de imágenes del rostro de las personas en tiempo real y clasificarlas manualmente guiándose por las definiciones según Ekman de la presencia de cada emoción y su relación con los datos faciales que captura el Kinect.

Su funcionamiento se descompone en las siguientes etapas:

1. Establecimiento de la comunicación con el sensor Kinect.
2. Detección y seguimiento de los puntos de interés del rostro de la persona.
3. Captura de la información facial de la persona.
4. Almacenamiento de la información.

Como expresa la metodología XP y se define según en el diagrama de flujo Figura 7, el proceso de desarrollo es un proceso iterativo e incremental que tiene que estar preparado para un ambiente cambiante. De esta forma si en algunas de las iteraciones los resultados no son los esperados se debe regresar al punto de partida de la funcionalidad y corregir los errores. La salida obtenida en la fase de Obtención de datos es probada en la definición de las primeras funcionalidades de la fase 2. De no obtenerse resultados favorables se procede a realizar un nuevo diseño de la funcionalidad encargada de capturar la información facial de la persona.

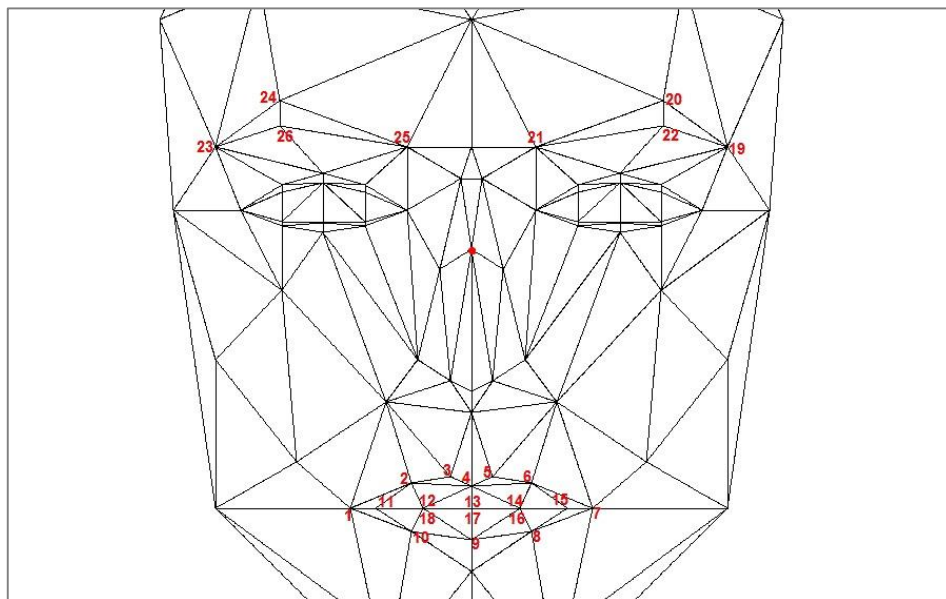
### 2.3.1 DISEÑO DE LA CAPTURA DE LA INFORMACIÓN FACIAL DE LA PERSONA

La información facial seleccionada del conjunto de datos que brinda el Kinect, son 26 puntos que indican la posición de determinados músculos del rostro en la forma (x; y). En la Tabla 4 se describe la relación que tienen estos con las AU de Ekman.

Tabla 4. Correspondencia entre las emociones y los puntos reconocidos por el Kinect.

Emoción a clasificar	AU correspondiente	Puntos del Kinect
Felicidad	AU6 Levantamiento de mejillas.	No identifica este movimiento.
	AU12 Tiramiento Labial esquinual	18 puntos (1 al 18) Ver Figura 8.
Tristeza	AU1 Levantamiento interior de cejas.	8 puntos (19 al 26) Ver Figura 8.
	AU4 Bajar cejas.	
	AU 15 Depresión labial esquinual.	18 puntos (1 al 18) Ver Figura 8.

Figura 8. Puntos re-enumerados como propuesta a utilizar.



Tal y como se obtienen estos valores no son adecuados para lograr una clasificación correcta ya que pueden no seguir igual distribución y proporciones generales ajustables a una escala en común.

Se realiza un pre-procesamiento de los datos para lograr una generalización y de esta forma obtener mejores resultados al modelar. La fórmula de distancia que se utiliza es la Euclidiana y el punto de referencia seleccionado es el del centro de la nariz, señalado en la *Figura 8*.

Paso 1: se toma un punto de referencia hacia el cual se calcula la distancia desde cada uno de los 26 puntos.

- (1) Distancia entre el punto  $P_i$  donde  $0 < i < 27$  y el punto de referencia  $P_r$ :

$$d_E(P_i, P_r) = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2}$$

Paso 2: se busca el menor y mayor punto reconocido del rostro, para calcular la máxima distancia entre puntos del rostro.

(2) Distancia entre el mayor ( $P_{max}$ ) y el menor punto ( $P_{min}$ ) del cuadrante:

$$d_E(P_{max}, P_{min}) = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}$$

Paso 3: para llevarlo a la escala [0; 1], se dividen las distancias  $d_E(P_i, P_r)$  y  $d_E(P_{max}, P_{min})$  para cada  $i$  de la lista de puntos.

La salida es un archivo de relación de atributos (.ARFF<sup>3</sup>) con los datos pre-procesados y las clasificaciones correspondientes (ver Anexo 1).

Luego de realizarse las pruebas a los datos capturados (ver detalles en el epígrafe 3.2), los resultados no son favorables y se procede a realizar otra variante del diseño de la Obtención de datos.

### 2.3.2 VARIANTE DE DISEÑO DE LA OBTENCIÓN DE DATOS DE ENTRENAMIENTO.

Como se describe en el Capítulo 1, el dispositivo Kinect es capaz de reconocer 6 Unidades de Animación, estas brindan información sobre la apariencia facial en tiempo real. Se pudo observar que las Unidades de Animación y las AU definidas por Ekman, guardan relación en cuanto a la definición de los movimientos faciales que deben experimentar las personas para poder afirmar la presencia o no de alguna emoción.

<sup>3</sup> (Attribute-Relation File Format) Formato utilizado por el conjunto de herramientas WEKA.

Figura 9. Relación entre las AU de Ekman y las Unidades de Animación del Kinect, emoción: Felicidad.

Emoción	Unidades de Acción (Ekman)	Unidades de Animación (Kinect)
Felicidad	AU6 Levantamiento de mejillas.	NR
	AU12 Tiramiento labial esquinal.	
		Unidad de Animación 1,2,3,5

Figura 10. Relación entre las AU de Ekman y las Unidades de Animación del Kinect, emoción: Tristeza.

Emoción	Unidades de Acción (Ekman)	Unidades de Animación (Kinect)
Tristeza	AU1 Levantamiento interior de cejas.	Unidad de Animación 4 y 6.
	AU4 Bajar cejas.	
	AU15 Depresión labial esquinal.	Unidad de Animación 1 y 5.

La salida para este diseño es una base de datos con los siguientes elementos:

1. Una imagen RGB de la captura clasificada.
2. Un archivo de texto con los 121 puntos de la forma (x; y) del rostro de la persona.
3. Las Unidades de Animación con sus respectivos valores y clasificación, en el formato (.ARFF). Este último es el que se utiliza para generar el modelo de clasificación en la segunda fase de desarrollo de la aplicación.

## 2.4 SELECCIÓN DEL MODELO DE CLASIFICACIÓN.

Esta fase es la encargada de evaluar el comportamiento de los algoritmos implementados en WEKA con el conjunto de datos de entrenamiento capturados en la fase anterior. Se procede a la obtención de un modelo

para la clasificación automática de nuevos casos, que a su vez será la salida de esta fase. Para seleccionar un modelo que sea capaz de generalizar y adaptarse al problema a resolver es necesario seguir las siguientes etapas:

1. Pre-procesamiento de los datos.
2. Configuración de parámetros del algoritmo de clasificación.
3. Configuración de parámetros del método de validación.
4. Generación del modelo de clasificación.
5. Evaluación y selección del modelo de clasificación.
6. Almacenamiento del modelo.

### **Pre-procesamiento de los datos:**

Esta es la primera parte por la que se debe pasar antes de realizar ninguna otra operación ya que engloba a todas aquellas técnicas de análisis de datos que permiten mejorar la calidad del conjunto predictor. De este modo las técnicas de extracción de conocimiento puedan obtener mayor y mejor información (mejor porcentaje de clasificación, reglas con más completitud). WEKA tiene integrados filtros que permiten realizar manipulaciones sobre los datos en dos niveles: atributos (verticales) e instancias (horizontales). Los filtros son de tipo:

- No supervisado (*unsupervised*): operaciones independientes del algoritmo de análisis posterior.
- Supervisados (*supervised*): operaciones en conjunción con algoritmos de clasificación para analizar su efecto (25) (26).

### **Métodos de validación:**

El resultado de aplicar el algoritmo de clasificación se evalúa comparando la clase predicha con la clase real de las instancias. Esta evaluación puede realizarse de diferentes modos:

- *Use training set*: evalúa el clasificador sobre el mismo conjunto que se construye el modelo predictivo para determinar el error. Por tanto, proporciona una estimación demasiado optimista del comportamiento del clasificador, al evaluarlo sobre el mismo conjunto sobre el que se hizo el modelo.
- *Supplied test set*: evaluación sobre un conjunto independiente. Permite cargar un conjunto nuevo de datos y sobre cada uno se realizará una predicción de clase para contar los errores.
- *Cross-Validation*: evaluación con validación cruzada. Es la más elaborada y costosa. Se realizan tantas evaluaciones como se indica en el parámetro *Folds*. Se dividen las instancias en tantas

carpetas como indica este parámetro y en cada evaluación se toman las instancias de cada carpeta como datos de *test* y el resto como datos de entrenamiento para construir el modelo. Los errores calculados son el promedio de todas las ejecuciones.

- *Percentage Split*: divide los datos en dos grupos, de acuerdo con el porcentaje indicado (%). El valor indicado es el porcentaje de instancias para construir el modelo, que a continuación es evaluado sobre las que se han dejado aparte. Cuando el número de instancias es suficientemente elevado, este método es suficiente para estimar con precisión las prestaciones del clasificador en el dominio (25).

## 2.5 CLASIFICACIÓN AUTOMÁTICA DE EMOCIONES EN LAS PERSONAS

En esta fase se procede a vincular el modelo de clasificación seleccionado anteriormente con las funcionalidades encargadas de la captura de datos de la primera fase para su reutilización y de esta forma obtener las clasificaciones en tiempo real. Para darle cumplimiento a lo antes descrito se definen las etapas siguientes:

1. Establecimiento de la comunicación con el sensor Kinect.
2. Detección y seguimiento de los puntos de interés del rostro de la persona.
3. Captura de la información facial de la persona.
4. Procesamiento de la información por el modelo de clasificación.
5. Resultado de la clasificación.

Las etapas 1, 2 y 3 se encuentran ya definidas en la fase de Obtención de los datos de entrenamiento ya que presentan características similares en la forma de capturar los datos.

## 2.6 INGENIERÍA DE SOFTWARE

### 2.6.1 ESPECIFICACIÓN DE REQUISITOS

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema como el control de un dispositivo, hacer un pedido o encontrar información. El proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se denomina ingeniería de requerimientos (27).



---

## REQUISITOS FUNCIONALES

---

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que este debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos también declaran explícitamente lo que el sistema no debe hacer (27).

Tabla 5: Requisitos Funcionales.

ID	Requisito Funcional
RF1	Establecer la comunicación con el sensor Kinect.
RF2	Visualizar captura de la cámara Kinect.
RF3	Detección y seguimiento de puntos en el rostro.
RF4	Visualizar malla del rostro.
RF5	Capturar imagen.
RF6	Mostrar valores de las Unidades de Animación.
RF7	Clasificar información.
RF8	Guardar la información.
RF9	Cargar modelo de clasificación.
RF10	Clasificar la emoción según el modelo utilizado.
RF11	Mostrar resultado.

---

## REQUISITOS NO FUNCIONALES

---

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. A menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema (27). A continuación, se especifican los requisitos no funcionales del sistema a desarrollarse.

### RnF 1. Usabilidad.

<Requisito de Usabilidad 1.1> Especificaciones de software:

- Sistema Operativo: *Microsoft Windows 7* o superior.
- Biblioteca y drivers del Kinect: *SDK "Kinect for Windows"*.

- Biblioteca del WEKA.

<Requisito de Usabilidad 1.2> Especificaciones del hardware:

- Procesador: *Intel Dual Core a 2.6 GHz.*
- Memoria: *2 GB de RAM.*
- Dispositivos de entrada adicionales: Kinect.

## RnF 2. Diseño.

<Requisito de Diseño 2.1> Lenguaje de programación:

- Para la implementación del sistema se debe usar el lenguaje C#.

### 2.6.2 FASE DE EXPLORACIÓN

En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo (24).

#### DESCRIPCIÓN DE REQUISITOS

En la programación extrema, todos los requerimientos se expresan como escenarios (llamados historias de usuario), los cuales se implementan directamente como una serie de tareas (27).

Las historias de usuario son la técnica utilizada en metodología XP para especificar los requisitos del software. Se trata de tarjetas en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Cada historia de usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (24).

A continuación, se muestran las historias de usuario definidas para esta aplicación:

Tabla 6. Historia de Usuario #1.

<b>Número:</b> 1	<b>Nombre del requisito:</b> Establecer la comunicación con el sensor Kinect.

<b>Programador:</b> Yaimara Fleites Reynaldo	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Puntos Estimados:</b> 1.0
<b>Descripción:</b> Cuando la aplicación se ejecute se debe establecer la comunicación del dispositivo Kinect con la aplicación verificando que haya algún dispositivo conectado.	
<b>Observaciones:</b>	

Tabla 7. Historia de Usuario #2.

<b>Número:</b> 2	<b>Nombre del requisito:</b> Visualizar captura de la cámara Kinect.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Media	<b>Puntos Estimados:</b> 1.0
<b>Descripción:</b> La interfaz debe ser capaz de mostrar la captura de imagen del dispositivo Kinect en tiempo real.	
<b>Observaciones:</b> De no observarse la captura, chequear que se esté cumpliendo la historia de usuario número 1.	

Tabla 8. Historia de Usuario #3.

<b>Número:</b> 3	<b>Nombre del requisito:</b> Detectar y seguir posición de los puntos en el rostro.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 1.
<b>Prioridad:</b> Alta.	<b>Puntos Estimados:</b> 1.0
<b>Descripción:</b> La aplicación debe ser capaz de detectar y darle seguimiento durante toda la captura a la posición de los puntos del rostro.	
<b>Observaciones:</b> En caso de no detectarse un rostro, deben de quedar bloqueadas las demás funcionalidades.	

Tabla 9. Historia de Usuario #4.

<b>Número:</b> 4	<b>Nombre del requisito:</b> Visualizar malla del rostro.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Media.	<b>Puntos Estimados:</b> 0.5
<b>Descripción:</b> La aplicación debe ser capaz de dibujar y mostrar la malla de los puntos del rostro.	
<b>Observaciones:</b> El usuario debe poder elegir si desea que se muestre la malla o no.	

Tabla 10. Historia de Usuario #5.

<b>Número:</b> 5	<b>Nombre del requisito:</b> Capturar imagen.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Media.	<b>Puntos Estimados:</b> 1.0
<b>Descripción:</b> El sistema debe ser capaz de pausar la captura en el momento que el usuario lo indique.	
<b>Observaciones:</b> Al pausar la captura deben de congelarse toda la información obtenida por el Kinect en ese instante de tiempo. Además debe brindar también la posibilidad de continuar con la captura.	

Tabla 11. Historia de Usuario #6.

<b>Número:</b> 6	<b>Nombre del requisito:</b> Mostrar valores de las Unidades de Animación.

<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 2						
<b>Prioridad:</b> Media.	<b>Puntos Estimados:</b> 1.0						
<b>Descripción:</b> El sistema debe ser capaz de mostrar los valores que adquieren las Unidades de Animación en tiempo real.							
<b>Observaciones:</b>							
<table border="1"> <tr><td><i>AU0: 1</i></td></tr> <tr><td><i>AU1: 0.3920485</i></td></tr> <tr><td><i>AU2: 0.7384062</i></td></tr> <tr><td><i>AU3: 0.4801158</i></td></tr> <tr><td><i>AU4: -0.6341062</i></td></tr> <tr><td><i>AU5: 0.01779734</i></td></tr> </table>		<i>AU0: 1</i>	<i>AU1: 0.3920485</i>	<i>AU2: 0.7384062</i>	<i>AU3: 0.4801158</i>	<i>AU4: -0.6341062</i>	<i>AU5: 0.01779734</i>
<i>AU0: 1</i>							
<i>AU1: 0.3920485</i>							
<i>AU2: 0.7384062</i>							
<i>AU3: 0.4801158</i>							
<i>AU4: -0.6341062</i>							
<i>AU5: 0.01779734</i>							

Tabla 12. Historia de Usuario #7.

<b>Número:</b> 7	<b>Nombre del requisito:</b> Clasificar información.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Alta.	<b>Puntos Estimados:</b> 0.5
<b>Descripción:</b> La aplicación debe brindarle al usuario la posibilidad de clasificar en Felicidad, Tristeza o Neutro la captura de la imagen realizada por el propio usuario.	
<b>Observaciones:</b> Solo se podrá clasificar una vez que se haya capturado o congelado una imagen en específico.	

Tabla 13. Historia de Usuario #8.

<b>Número:</b> 8	<b>Nombre del requisito:</b> Guardar información.

<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 3
<b>Prioridad:</b> Alta.	<b>Puntos Estimados:</b> 2.0
<b>Descripción:</b> La aplicación debe ser capaz de guardar la siguiente información capturada por el Kinect: <ul style="list-style-type: none"> <li>• Imagen RGB con extensión .jpg.</li> <li>• Puntos del rostro con extensión .txt.</li> <li>• Unidades de Animación + Clasificación con extensión .arff.</li> </ul>	
<b>Observaciones:</b> La funcionalidad no debe estar habilitada mientras el usuario no haya clasificado la captura.	

Tabla 14. Historia de Usuario #9.


<b>Número:</b> 9	<b>Nombre del requisito:</b> Cargar modelo de clasificación.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 4
<b>Prioridad:</b> Alta.	<b>Puntos Estimados:</b> 1.5
<b>Descripción:</b> La aplicación debe ser capaz de cargar el modelo de clasificación.	
<b>Observaciones:</b> Tener en cuenta que los modelos pueden variar y no ser siempre el mismo, pero solo se trabajará con un modelo a la vez.	

Tabla 15. Historia de Usuario #10.

<b>Número:</b> 10	<b>Nombre del requisito:</b> Clasificar la emoción según el modelo utilizado.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 4
<b>Prioridad:</b> Alta.	<b>Puntos Estimados:</b> 1.0

<b>Descripción:</b> La aplicación debe ser capaz de clasificar las nuevas entradas de datos según el modelo de clasificación desarrollado.
<b>Observaciones:</b> La clasificación resultante debe ser la moda de 11 clasificaciones realizadas cada 1 segundos.

Tabla 16. Historia de Usuario #11.

<b>Número:</b> 11	<b>Nombre del requisito:</b> Mostrar resultado.
<b>Programador:</b> Yaimara Fleites Reynaldo.	<b>Iteración Asignada:</b> 4
<b>Prioridad:</b> Media.	<b>Puntos Estimados:</b> 0.5
<b>Descripción:</b> La aplicación debe ser capaz de mostrar en pantalla el resultado de la clasificación de la nueva entrada de datos en tiempo real.	
<b>Observaciones:</b>	
	

El tratamiento de las historias de usuario es muy dinámico y flexible, en cualquier momento las historias de usuario pueden reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas.

### 2.6.3 FASE DE PLANIFICACIÓN

En esta fase el cliente establece la prioridad de cada historia de usuario y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según

alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación (24).

## ESTIMACIÓN DE ESFUERZO

Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la "velocidad" de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración (24).

Tabla 17. Estimación de esfuerzo por Historia de Usuario.

Iteración	Historias de Usuario	Puntos estimados (semanas)
1	1 Establecer la comunicación con el sensor Kinect.	1.0
	2 Visualizar captura de la cámara Kinect.	1.0
	3 Detección y seguimiento de puntos en el rostro.	1.0
2	4 Visualizar malla del rostro.	0.5
	5 Capturar imagen.	1.0
	6 Mostrar valores de las Unidades de Animación.	1.0
3	7 Clasificar información.	0.5
	8 Guardar la información.	2.0
4	9 Cargar modelo de clasificación.	1.5
	10 Clasificar la emoción según el modelo utilizado.	1.0
	11 Mostrar resultado.	0.5
<b>Total</b>		<b>11.0</b>

## PLAN DE ITERACIONES

El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura. Al final de la última



iteración el sistema estará listo para entrar en producción (24). A continuación se describen las iteraciones definidas para el desarrollo de la aplicación:

**Iteración #1:** Se implementan las historias de usuario 1, 2 y 3, las cuales realizan funcionalidades básicas para definir la arquitectura de la aplicación. Al culminar esta iteración se realizará la entrega de la versión 1.0 al cliente para evaluar los resultados parciales.

**Iteración #2:** En esta iteración se rectificarán los errores encontrados en la iteración anterior y se le darán cumplimiento a las historias de usuario 4, 5 y 6 que definen funcionalidades específicas para la resolución del problema.

**Iteración #3:** En esta iteración se le da cumplimiento a las historias de usuario 7 y 8 que garantizan la etapa final de la extracción de datos. Luego de esta iteración se realizará la entrega de la versión 1.1 de la aplicación completándose de esta forma el Módulo de Captura de Datos.

**Iteración #4:** En esta iteración se rectificarán los errores detectados en la iteración anterior y se desarrollarán las historias de usuario 9, 10 y 11, siendo estas las referentes a la clasificación de emociones en tiempo real por lo cual tienen gran relevancia. Luego de culminada esta iteración se obtiene el producto final en su versión 1.2.

---

## PLAN DE ENTREGA

---

A partir de las iteraciones definidas anteriormente se establece un plan de entrega que tiene como objetivo fijar las fechas y el orden de la entrega de versiones, ver Tabla 18.

Tabla 18. Plan de entregas.

Iteración	Versión	Fecha de entrega
1	1.0	27 de febrero del 2016
2,3	1.1	22 de abril del 2016
4	1.2	3 de junio del 2016

### 2.6.4 FASE DE DISEÑO

---

El diseño es una actividad que tiene que ver con la toma de decisiones importantes, con frecuencia de naturaleza estructural. Comparte con la programación el objetivo de abstraer una representación de la

información y de las secuencias de procesamiento, pero en los extremos el grado de detalle es muy distinto. El diseño elabora representaciones coherentes y bien planeadas de programas, que se concentran en las relaciones de las partes en el nivel más alto y en las operaciones lógicas involucradas en los niveles bajos (28).

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. En cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos (24).

---

## ARQUITECTURA

---

El diseño arquitectónico representa la estructura de los datos y de los componentes del programa que se requieren para construir un sistema basado en computadora. Considera el estilo de arquitectura que adoptará el sistema, la estructura y las propiedades de los componentes que lo constituyen y las interrelaciones que ocurren entre sus componentes arquitectónicos.

El diseño de la arquitectura comienza con el diseño de los datos y continúa con la obtención de una o más representaciones de la estructura arquitectónica del sistema. Se analizan alternativas de estilos o patrones arquitectónicos para llegar a la estructura más adecuada para los requerimientos del usuario y para los atributos de calidad (28).

Para el desarrollo de la aplicación, luego de un estudio realizado a los distintos diseños de arquitectura y atendiendo a sus características se utilizará la arquitectura basada en capas, la cual se describe a continuación.

### *Arquitectura N-capas*

En esta arquitectura se define un número de capas diferentes; cada una ejecuta operaciones que se aproximan progresivamente al conjunto de instrucciones de máquina. En la capa externa, los componentes atienden las operaciones de la interfaz de usuario. En la interna, los componentes realizan la interfaz con el sistema operativo. Las capas intermedias proveen servicios de utilerías y funciones de software de aplicación (28). Ver Figura 11.

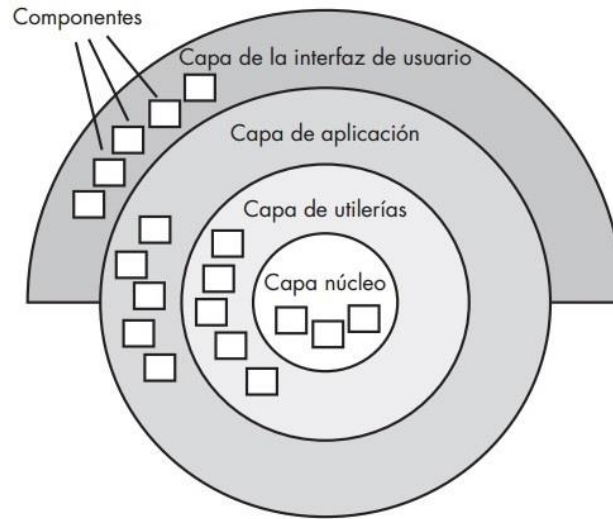


Figura 11. Arquitectura por capas.

La aproximación por capas soporta el desarrollo incremental de sistemas. A medida que se desarrolla una capa, algunos de los servicios proporcionados por esa capa pueden estar disponibles para los usuarios. Esta arquitectura también soporta bien los cambios y es portable. En la medida en que la interfaz permanezca sin cambios, una capa puede reemplazarse por otra capa equivalente. Además, cuando las interfaces de la capa cambian o se añaden nuevas facilidades a una capa, solamente se ve afectada la capa adyacente (27).

Para el diseño de la aplicación se adopta una arquitectura de tres capas como se muestra en la *Figura 12*.

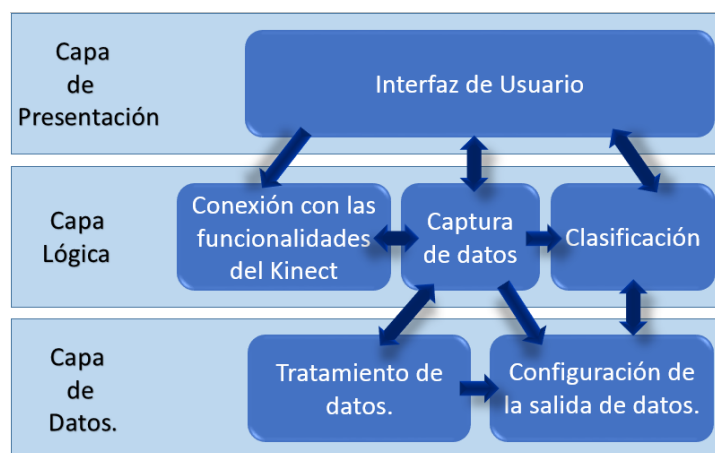


Figura 12. Diseño de la Arquitectura de 3 capas del sistema.

**Capa de presentación:** Es la capa que facilita la interfaz que se comunica con el usuario abstrayéndolo del proceso interno de la aplicación.

**Capa lógica:** En esta capa se controla el proceso de captura de la información proveniente del dispositivo Kinect para la generación de los datos su envío a la capa de datos. Además, se realizan las funcionalidades propias del proceso de clasificación de emociones.

**Capa de datos:** En esta capa se enlazan las funciones proporcionadas por la SDK del Kinect para poder trabajar con ellas. De esta forma pueden ser obtenidos los datos capturados por el Kinect y almacenados según los requerimientos planteados.

---

## PATRONES DE DISEÑO

---

El diseño basado en patrones crea una aplicación nueva, encontrando un conjunto de soluciones comprobadas para un conjunto de problemas delineado con claridad. Cada problema y su solución está descrito por un patrón de diseño catalogado y analizado por otros ingenieros de software que han encontrado el problema e implantado su solución cuando diseñaban otras aplicaciones. Cada patrón de diseño provee un enfoque demostrado para una parte del problema que debe resolverse (28).

Los patrones GRASP (*General Responsibility Assignment Software Patterns*, por sus siglas en inglés) nos brindan unos principios generales para asignar responsabilidades y se utiliza sobre todo en la realización de diagramas de interacción. Los patrones GOF (*Gang of Four*) han tenido un gran éxito en el mundo de la informática ya que su utilización constituye una de las buenas prácticas de diseño de clases a seguir. A continuación se exponen los patrones a utilizar en el desarrollo de la aplicación:

**Experto:** Se encarga de asignar una responsabilidad al experto en información – la clase que tiene la información necesaria para la realización de la asignación. Indica que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria sobre este. Este patrón fue utilizado en todas las clases.

```

public class OutputManager
{
    2 references
    public List<Point> facePoints { get; set; }
    3 references
    public List<float> AUvalues { get; set; }
    2 references
    public WriteableBitmap image { get; set; }
    2 references
    public string clasif { get; set; }
    1 reference
    public OutputManager() { }
    1 reference
    void savePoints(string filename) [...]
    1 reference
    void saveAU(string filename) [...]
    1 reference
    void saveImage(string filename) [...]
    1 reference
    public void generateFile() [...]
    1 reference
    public void alert() [...]
}

```

Figura 13. Ejemplo de utilización del patrón Experto.

**Creador:** Determina quién debe ser responsable en la creación de una nueva instancia de una clase. La clase MainWindows.xaml.cs tiene la responsabilidad de crear las estructuras y objetos que contendrán la información para su posterior análisis.

```

public partial class MainWindow : Window
{
    (...)
    private void Button_GuardarInfo_Click(object sender, RoutedEventArgs e)
    {
        List<Point> facePointsList = faceTrackingViewer.FacePoints;
        List<float> AUvalues = faceTrackingViewer.AUvalues;

        OutputManager outputManager = new OutputManager
        {
            facePoints = facePointsList,
            AUvalues = AUvalues,
            image = colorImageWritableBitmap,
            clasif = this.clasif
        };
        outputManager.generateFile();
    }
}

```

Figura 14. Ejemplo de utilización del patrón Creador.

**Alta cohesión:** Es la encargada de que la información que almacena cada clase tenga coherencia y esté vinculada con la misma. Este patrón fue utilizado en todas las clases.

**Bajo acoplamiento:** Permite tener solo las relaciones necesarias y suficientes entre las clases disminuyendo la dependencia entre ellas. Este patrón se tuvo en cuenta para las relaciones entre todas las clases.

**Singleton:** Este patrón está diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto. Consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella. Este patrón es utilizado en la clase `ClassifyInstances` para lograr que la funcionalidad de cargar el modelo de clasificación, puesto que adquiere un valor global único en el sistema, se ejecute solo una vez.

```
class ClassifyInstances
{
    private static volatile ClassifyInstances instance;

    Classifier smo;

    1 reference
    private ClassifyInstances()
    {
        smo = (Classifier)weka.core.SerializationHelper.read("datas/SMO.model");
    }

    2 references
    public static ClassifyInstances getInstance()
    {
        if (instance == null)
            instance = new ClassifyInstances();
        return instance;
    }
}
```

Figura 15. Ejemplo de utilización del patrón Singleton.

---

## TARJETAS CRC

---

El modelado Clase-Responsabilidad-Colaborador (CRC), proporciona una manera sencilla de identificación y organización de las clases que son relevantes para los requerimientos de un sistema o producto.

Un modelo CRC en realidad es un conjunto de tarjetas índice estándar que representan clases. Las tarjetas se dividen en tres secciones. En la parte superior de la tarjeta se escribe el nombre de la clase, en la parte izquierda del cuerpo se enlistan las responsabilidades de la clase y en la derecha, los colaboradores (28).

Tabla 19. Tarjeta CRC #1.

Tarjeta CRC	
<b>Clase:</b> MainWindow.xaml.cs	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Permitir el acceso a las funcionalidades de la aplicación.</li> <li>• Permite crear las estructuras de datos y las instancias de objetos para su posterior uso.</li> <li>• Mostrar mapa de bits.</li> <li>• Capturar mapa de bits.</li> <li>• Clasificar datos.</li> <li>• Mostrar la clasificación de emociones en tiempo real.</li> </ul>	OutputManager.cs FaceTrackingViewer.xaml.cs ClassifyInstances.cs

Tabla 20. Tarjeta CRC #2.

Tarjeta CRC	
<b>Clase:</b> FaceTrackingViewer.xaml.cs	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Establecer la comunicación con el Kinect.</li> <li>• Obtener información facial.</li> </ul>	

Tabla 21. Tarjeta CRC #3.

Tarjeta CRC	
<b>Clase:</b> OutputManager.cs	

Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Guardar imagen RGB.</li> <li>• Guardar puntos faciales.</li> <li>• Guardar Unidades de Animación.</li> </ul>	MainWindows.xaml.cs

Tabla 22. Tarjeta CRC #4.

Tarjeta CRC	
<b>Clase:</b> ClassifyInstances.cs	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Cargar el modelo de clasificación.</li> <li>• Cargar instancias a clasificar.</li> <li>• Pre-procesar los datos.</li> <li>• Clasificar las predicciones.</li> </ul>	MainWindows.xaml.cs OutputManager.cs

La metodología XP propone el uso de tarjetas CRC en lugar de una serie de complejos diagramas que probablemente tomen más tiempo y sean menos instructivos.

## CONCLUSIONES DEL CAPÍTULO

- El diseño realizado cumple con los procesos descritos para lograr el PR.
- Los puntos reconocidos por el Kinect pueden ser utilizados para definir la presencia o no de una Unidad de Acción.
- Las Unidades de Acción definidas por Ekman están relacionadas con las Unidades de Animación generadas por el Kinect.
- El tipo de clasificación a efectuar por la naturaleza del problema es la clasificación supervisada.



## CAPÍTULO 3. ANÁLISIS Y RESULTADOS

### 3.1 INTRODUCCIÓN

En este capítulo se realiza una valoración de los resultados obtenidos durante el proceso de implementación y de mejoras. Se describe el proceso de codificación de la aplicación. Se establecen los estándares de codificación, así como las estructuras de los archivos generados por el sistema. Además, se muestran las pruebas de aceptación realizadas para comprobar el cumplimiento de los requisitos establecidos por el cliente.

### 3.2 SELECCIÓN DEL MODELO DE CLASIFICACIÓN

Luego de generados los archivos .ARFF en la fase de Obtención de los datos de entrenamiento, se procede a aplicar los métodos de pre-procesamiento a los datos y se evalúa el comportamiento de los algoritmos de clasificación para los mismos.

Inicialmente a los datos se les aplica el filtro *Normalize* para la normalización de todos los atributos con valores de escala igual a 2.0 y traslación -1.0. Además, se utiliza como método de validación *Cross-Validation* de 10 hojas.

La Tabla 23 muestra algunos de los resultados obtenidos, en cuanto a porcentaje de instancias correctamente clasificadas por la captura de datos correspondiente a los 26 puntos reconocidos por el Kinect.

Tabla 23. Porcientos de instancias correctamente clasificadas con puntos faciales como datos.

Datos\Algoritmo	Puntos faciales
trees.J48	34.62%
trees.DecisionStump	38.46%
rules.DecisionTable	38.46%
rules.PART	38.46%
meta.MultiClassClassifier	46.15%
lazy.IBK	42.31%
lazy.LWL	42.31%
functions.SimpleLogistic	46.15%

functions.SMO	<b>50%</b>
---------------	------------

Como se observa los resultados alcanzados por esta vía no son favorables, obteniéndose como promedio de instancias correctamente clasificadas un 42%. El mejor resultado lo alcanza el algoritmo SMO (Sequential Minimal Optimization) con un 50 % de instancias correctamente clasificadas.

Al comprobar los bajos porcentos de clasificaciones correctas, obtenidos con los algoritmos, se puede afirmar que el problema radica en los datos capturados. Por esta razón se rediseña una variante de la Obtención de los datos de entrenamiento expuesta en el epígrafe 2.3.2.).

Para probar la variante de datos, se aplican los mismos filtros que se le aplicaron al juego de datos anterior en la herramienta WEKA, obteniéndose los resultados que muestra la Tabla 24.

Tabla 24. Porcientos de instancias correctamente clasificadas con la variante de Datos.

Datos\Algoritmo	Unidades de Animación
trees.J48	43.33%
trees.DecisionStump	<b>70%</b>
rules.DecisionTable	<b>70%</b>
rules.PART	46.67%
meta.MultiClassClassifier	46.67%
lazy.IBK	53.33%
lazy.LWL	66.67%
functions.SimpleLogistic	66.67%
functions.SMO	<b>70%</b>

Como se observa en la Tabla 24 con esta variante se logró alcanzar hasta un 70% de instancias correctamente clasificadas con varios algoritmos. Además, coincide nuevamente que el SMO es de los que obtiene mejores resultados.

### 3.2.1 RESULTADOS DE LA COMPARACIÓN ENTRE AMBOS DISEÑOS

Luego de analizar por separado ambos casos se pudo constatar que para los mismos algoritmos analizados, se obtuvieron indicadores más elevados con la variante de diseño la Obtención de datos de entrenamiento. Ver Figura 16.

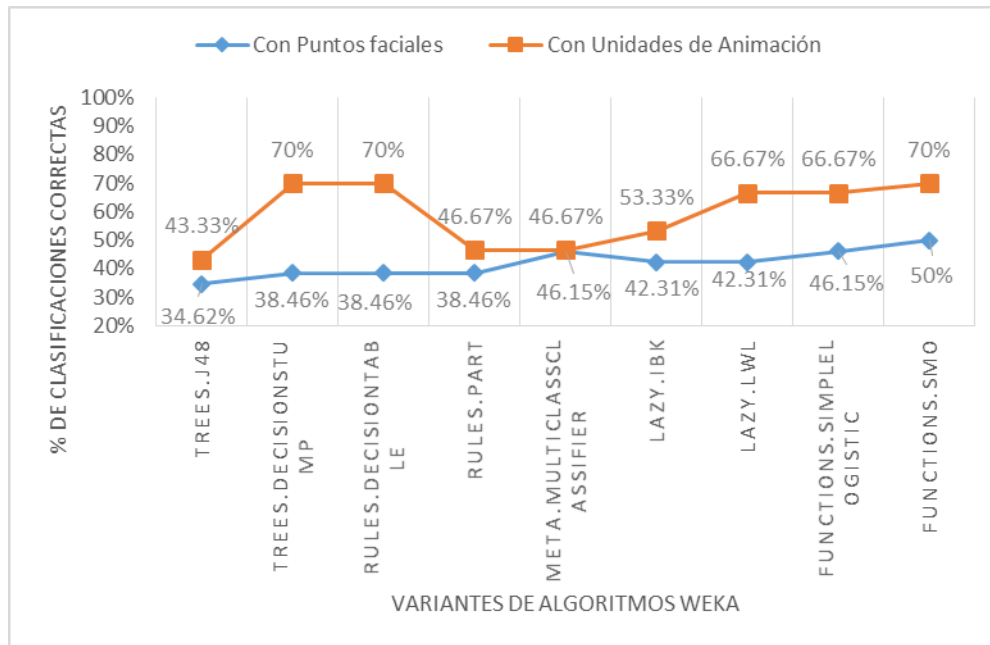


Figura 16. Comportamiento de los algoritmos en WEKA según el juego de datos.

Por lo anteriormente demostrado se decide proseguir con la tercera y última fase de desarrollo de la aplicación, Selección del modelo de clasificación, en la que se utilizará la variante de diseño descrita en el Capítulo 2 del documento.

### 3.2.2 MEJORAS IMPLEMENTADAS

Con el objetivo de mejorar los resultados alcanzados hasta el momento se procede a aplicarle otros filtros al conjunto de datos en la herramienta WEKA.

#### SELECCIÓN DE ATRIBUTOS

Es aconsejable tratar la selección de atributos (eliminación de atributos redundantes e irrelevantes) ya que si existiera un número excesivo de atributos, esto puede hacer que el modelo sea demasiado complejo y se produzca *overfitting*<sup>4</sup>.

<sup>4</sup> Sobreajuste.

Para ello se seleccionó en la pestaña *attribute selection* el método de búsqueda *Ranker* y como método de evaluación *InfoGain* y el método de test para evaluar los atributos será de *crossvalidation* de 5 hojas. Esto hará que la evaluación sea 5 veces más lenta, pero más precisa.

average merit	average rank	attribute
0.747 +- 0.012	1 +- 0	1 valor1
0.397 +- 0.206	2.4 +- 0.49	5 valor5
0.401 +- 0.211	3.2 +- 1.47	3 valor3
0 +- 0	4.2 +- 0.4	4 valor4
0 +- 0	4.6 +- 0.8	2 valor2
0 +- 0	5.6 +- 0.8	6 valor6

Figura 17. Resultado de la selección de atributos en WEKA.

*Ranker* nos ordenaba los atributos y ha considerado y ha considerado que los atributos 1, 5 y 3 son los mejores.

Se procede a probar el algoritmo SMO con el filtro de atributos aplicado y el porcentaje de clasificaciones correctas aumentó en un 3.33%.

## FILTROS DE PRE-PROCESAMIENTO

Al analizar las matrices de confusión arrojadas en cada prueba, se pudo observar que la clasificación con menor porcentaje de instancias correctas era la tristeza ya que sus valores tienen un gran parecido a los del rostro neutro, lo que afecta el resultado final.

```

=== Confusion Matrix ===
  a  b  c  <-- classified as
10  1  0 | a = f
 3  0  5 | b = t
 0  0 11 | c = n

```

Figura 18. Matriz de Confusión para el algoritmo SMO.

Para solucionar este problema, se tiene en cuenta que ya para el reconocimiento de las Unidades de Animación el Kinect realiza la inferencia de un rostro neutro y se decide aplicar el filtro *RemoveWithValues* para eliminar todas las instancias que fueron clasificadas de neutro.

Culminado este paso se procede a probar nuevamente los algoritmos implementados en WEKA y se logra obtener hasta un 84.21% de instancias clasificadas correctamente con el algoritmo SMO, como se muestra en la Figura 19.

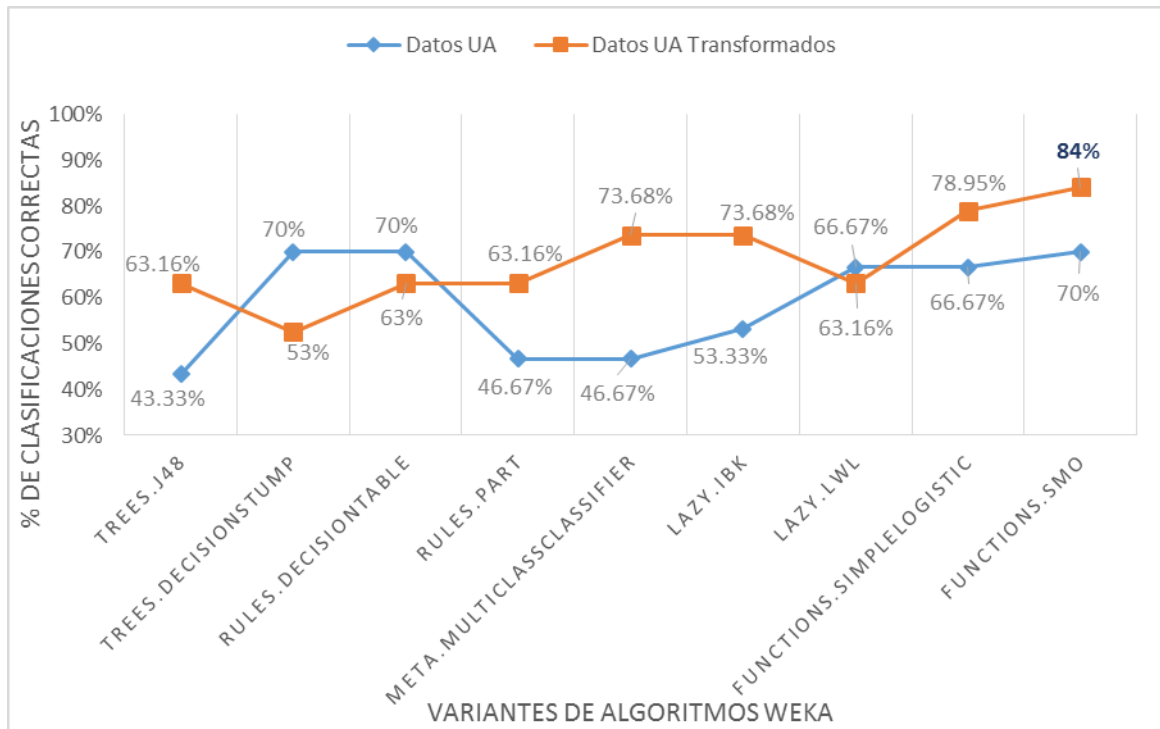


Figura 19. Comportamiento de los algoritmos en WEKA según los datos de las Unidades de Animación y sus modificaciones.

### 3.2.3 ALGORITMO DE OPTIMIZACIÓN SECUENCIAL MÍNIMA (SMO)

El algoritmo SMO Implementa el algoritmo de optimización secuencial mínima de John Platt<sup>5</sup> para la formación de un clasificador de vectores de soporte. Esta aplicación sustituye a nivel global todos los valores perdidos y transforma atributos nominales en unos binarios. También normaliza todos los atributos por defecto. En ese caso, los coeficientes en la salida se basan en los datos normalizados, no los datos

<sup>5</sup> Doctor en Ciencias Computacionales. Departamento de Investigación de la Corporación Google.

originales, esto es importante para la interpretación del clasificador. Este clasificador es una variante de las Máquinas de Soporte Vectorial (29).

## 3.3 CODIFICACIÓN

### 3.3.1 ESTÁNDARES DE CODIFICACIÓN

La metodología XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación. Estos mantienen el código legible para los miembros del equipo, facilitando los cambios.

Como estándar de codificación de la aplicación se adoptaron, cumpliendo con las siguientes restricciones:

#### NOMBRES

1. Los nombres de los archivos se escribirán siempre en mayúscula y en idioma inglés. En caso de que los nombres sean compuestos la próxima palabra deberá comenzar con letra inicial mayúscula.

Ejemplo: OutputManager.cs, MainWindows.xaml.

2. Los identificadores para los espacios de nombre y las clases siempre serán sustantivos y en idioma inglés. Comenzarán además con la letra inicial mayúscula. El resto del nombre será en minúscula.

Ejemplo: `public class OutputManager`

```
{  
};
```

Si el identificador de un espacio de nombre o de una clase es compuesto, se escribirá la letra inicial de cada palabra en mayúscula.

3. Los atributos de clases, parámetros y variables locales se escribirán todos comenzando con minúscula, de ser una palabra compuesta se escribirá la primera palabra en minúscula y a partir de ese momento cada nueva palabra se escribirá con letra inicial mayúscula.

Ejemplo: `public partial class MainWindow : Window`

```
{
```

```
private bool captureImage;

Private string clasific;

private List<int> listPoints;

};
```

4. Los métodos siguen el mismo convenio de las variables.

Ejemplo: `public class OutputManager`

```
{
    void savePoints(string filename){}

    void saveAU(string filename){}

    void saveImage(string filename){}
};
```

Nota: Todos los nombres serán simples y descriptivos evitando poner nombres de un solo carácter, excepto para variables de índices temporales.

---

## INDENTACIÓN

---

La indentación será de cuatro espacios, utilizándose la tabulación. No poseerán indentación las llaves asociadas a los espacios de nombre, definición de clases, implementación de métodos, condicionales y bucles.

---

## LÍNEAS Y ESPACIOS EN BLANCO

---

Se insertará una línea en blanco entre las definiciones de clases y métodos. Los operadores binarios estarán separados de sus operandos por un espacio en blanco, excepto en el caso de los operadores unitarios de incremento (++) y decremento (--). De igual forma serán separados los argumentos de un método, insertando un espacio después de la coma.

## 3.4 ARCHIVOS GENERADOS

En la Obtención de datos de entrenamiento se generan tres archivos: una imagen RGB de la captura realizada, los 121 puntos faciales reconocidos por el Kinect, referentes a esta y los valores de las seis Unidades de Animación más la clasificación.

### 3.4.1 ESTRUCTURA DE LOS FICHEROS

Para cada clasificación será creada una carpeta que contenga los tres archivos correspondientes a esta. El nombre de la carpeta estará compuesto por la fecha exacta como se muestra a continuación:

dd-mm-yyyy hh mm ss

Siendo:

- dd = Día
- mm = Mes
- yyyy = Año
- hh = hora
- mm = minuto
- ss = segundo

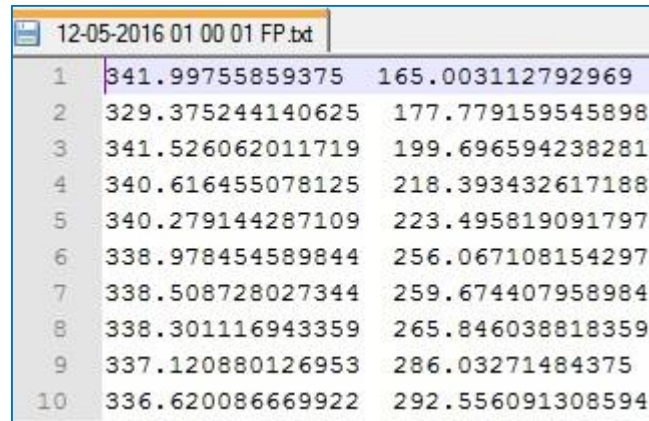
Cada uno de los tres archivos guardados dentro de la carpeta contendrá también la misma estructura de la fecha pero además en dependencia del tipo de archivo al final se les identificará con un identificador del contenido como se muestra en la tabla:

Tabla 25. Estructura del nombre de los archivos generados por el Módulo de Captura de Datos.

Información contenida	Estructura del nombre de los archivos		
	Fecha	Identificador	Extensión
Imagen RGB de la captura realizada.	dd-mm-yyyy hh mm ss	IMG	.jpg
121 puntos faciales.	dd-mm-yyyy hh mm ss	FP	.txt
Unidades de Animación + clasificación.	dd-mm-yyyy hh mm ss	AU	.arff
<b>Nombre del archivo = Fecha + Identificador + Extensión</b>			



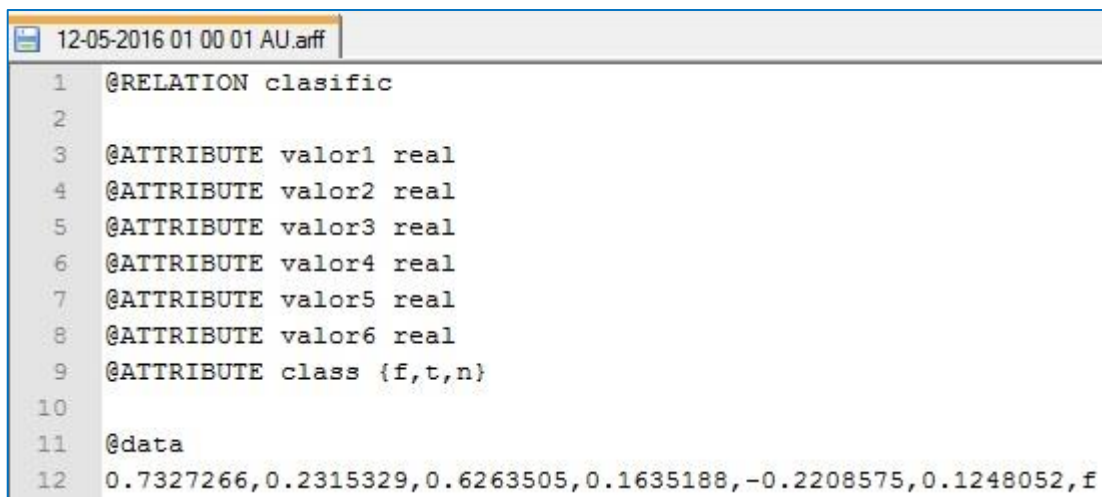
Los 121 puntos 2D reconocidos por el Kinect son guardados en un archivo de texto de la forma: “Xi (espacio) Yi”, seguido por un salto de línea entre cada punto. En la Figura 20 se muestra un ejemplo para los diez primeros puntos.



Line	X	Y
1	341.99755859375	165.003112792969
2	329.375244140625	177.779159545898
3	341.526062011719	199.696594238281
4	340.616455078125	218.393432617188
5	340.279144287109	223.495819091797
6	338.978454589844	256.067108154297
7	338.508728027344	259.674407958984
8	338.301116943359	265.846038818359
9	337.120880126953	286.03271484375
10	336.620086669922	292.556091308594

Figura 20. Archivo de puntos faciales generado por la fase de Obtención de datos de entrenamiento.

Las Unidades de Animación por su parte se guardan en un archivo .ARFF para su posterior tratamiento en la herramienta WEKA. Está compuesto por una cabecera que contiene la relación de atributos, el tipo de atributos a guardar y los datos clasificados. La Figura 21 muestra un ejemplo para una captura clasificada de “felicidad”.



```
1 @RELATION clasific
2
3 @ATTRIBUTE valor1 real
4 @ATTRIBUTE valor2 real
5 @ATTRIBUTE valor3 real
6 @ATTRIBUTE valor4 real
7 @ATTRIBUTE valor5 real
8 @ATTRIBUTE valor6 real
9 @ATTRIBUTE class {f,t,n}
10
11 @data
12 0.7327266,0.2315329,0.6263505,0.1635188,-0.2208575,0.1248052,f
```

Figura 21. Archivo de .ARFF generado por la fase de Obtención de datos de entrenamiento.

### 3.5 PRUEBAS

Uno de los pilares de la XP es el proceso de pruebas. XP anima a probar constantemente tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñada por los programadores y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñadas por el cliente final (30).

Las pruebas de aceptación son una parte integral del desarrollo incremental tal como se practica por XP. Todas las historias de los usuarios son apoyados por pruebas de aceptación, que se definen por el propio cliente. Estas pruebas hacen frente a los temores de que el negocio se ha entendido mal (31).

Tabla 26. Caso de prueba de aceptación #1.

Caso de prueba de aceptación	
<b>Código:</b> HU1p1	<b>Historia de Usuario:</b> 1.
<b>Nombre:</b> Establecer la comunicación con el sensor Kinect.	
<b>Descripción:</b> Prueba si se establece correctamente la comunicación con el sensor Kinect.	
<b>Condición de ejecución:</b>	
<ul style="list-style-type: none"> <li>La conexión debe establecerse inmediatamente que se ejecute la aplicación.</li> </ul>	
<b>Paso de ejecución:</b>	
<ul style="list-style-type: none"> <li>(Se activa automáticamente cuando la aplicación es ejecutada por primera vez.)</li> </ul>	
<b>Resultados esperados:</b>	
<ul style="list-style-type: none"> <li>Si la conexión se estableció de forma correcta se deben de poder capturar las informaciones de imagen.</li> </ul>	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 27. Caso de prueba de aceptación #2.

Caso de prueba de aceptación	
<b>Código:</b> HU2p1	<b>Historia de Usuario:</b> 2

<b>Nombre:</b> Visualizar captura de la cámara Kinect.
<b>Descripción:</b> Prueba la funcionalidad mostrar la captura de la cámara Kinect.
<b>Condición de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la UH #1.</li> </ul>
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• (Se activa automáticamente cuando la aplicación es ejecutada por primera vez.)</li> </ul>
<b>Resultados esperados:</b> <ul style="list-style-type: none"> <li>• Visualización del video (<i>frames</i>) capturados por el sensor Kinect en la interfaz de la aplicación.</li> </ul>
<b>Evaluación de la prueba:</b> Satisfactorio.

Tabla 28. Caso de prueba de aceptación #3.

Caso de prueba de aceptación	
<b>Código:</b> HU3p1	<b>Historia de Usuario:</b> 3
<b>Nombre:</b> Detección y seguimiento de puntos en el rostro.	
<b>Descripción:</b> Prueba si se realiza correctamente por parte del Kinect la detección y seguimiento del rostro.	
<b>Condición de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #1.</li> <li>• Debe estar el rostro de una persona de frente al sensor a una distancia en el metro y los tres metros.</li> </ul>	
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• (Se activa automáticamente cuando la aplicación es ejecutada por primera vez y se actualiza a razón de 30 <i>frames</i> por segundo.)</li> </ul>	
<b>Resultados esperados:</b> <ul style="list-style-type: none"> <li>• Se debe poder capturar la información facial de la persona.</li> <li>• Se habilitan los botones “Mostrar malla” y “Capturar”.</li> </ul>	
<b>Evaluación de la prueba:</b> Satisfactorio. <ul style="list-style-type: none"> <li>• Debe tenerse en cuenta que a pesar de ser “Satisfactorio” el resultado pues cumple con las expectativas mínimas, el Kinect no fue capaz de reconocer la acción de la AU6 “Levantamiento de mejillas” y en algunas pocas ocasiones presentó problemas para reconocer gestos que el usuario estaba efectuando.</li> </ul>	

Tabla 29. Caso de prueba de aceptación #4.

Caso de prueba de aceptación	
<b>Código:</b> HU4p1	<b>Historia de Usuario:</b> 4

<b>Nombre:</b> Visualizar malla del rostro.
<b>Descripción:</b> Prueba la funcionalidad mostrar malla del rostro.
<b>Condición de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #3.</li> </ul>
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• Seleccionar la opción “Mostrar malla”.</li> </ul>
<b>Resultados esperados:</b> <ul style="list-style-type: none"> <li>• Se debe visualizar la malla correspondiente al rostro detectado por el Kinect.</li> </ul>
<b>Evaluación de la prueba:</b> Satisfactorio.

Tabla 30. Caso de prueba de aceptación #5.

Caso de prueba de aceptación	
<b>Código:</b> HU5p1	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Capturar imagen.	
<b>Descripción:</b> Prueba la funcionalidad capturar imagen.	
<b>Condición de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #3.</li> <li>• Debe estar activado el Modo de Clasificación Manual.</li> </ul>	
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• Seleccionar la opción “Capturar”.</li> </ul>	
<b>Resultados esperados:</b> <ul style="list-style-type: none"> <li>• Se debe congelar la imagen y los datos de ese instante de tiempo.</li> <li>• Se activa la funcionalidad de la Historia de Usuario #7.</li> </ul>	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 31. Caso de prueba de aceptación #6.

Caso de prueba de aceptación	
<b>Código:</b> HU6p1	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Mostrar valores de las Unidades de Animación.	
<b>Descripción:</b> Prueba la funcionalidad mostrar los valores de las Unidades de Animación.	
<b>Condición de ejecución:</b>	

<ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #3.</li> </ul>
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• (Se activa automáticamente cuando la aplicación es ejecutada por primera vez.)</li> </ul>
<b>Resultados esperados:</b> <ul style="list-style-type: none"> <li>• Se deben mostrar en la interfaz los identificadores y valores de las Unidades de Animación.</li> </ul>
<b>Evaluación de la prueba:</b> Satisfactorio.

Tabla 32. Caso de prueba de aceptación #7.

Caso de prueba de aceptación	
<b>Código:</b> HU7p1	<b>Historia de Usuario:</b> 7
<b>Nombre:</b> Clasificar información.	
<b>Descripción:</b> Prueba la funcionalidad clasificar los datos faciales.	
<b>Condición de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe estar activado el Modo de Clasificación Manual.</li> <li>• Debe estar activada la funcionalidad “Capturar”.</li> </ul>	
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• Marcar la clasificación deseada.</li> </ul>	
<b>Resultados esperados:</b> <ul style="list-style-type: none"> <li>• Debe marcarse el atributo seleccionado.</li> </ul>	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 33. Caso de prueba de aceptación #8.

Caso de prueba de aceptación	
<b>Código:</b> HU8p1	<b>Historia de Usuario:</b> 8
<b>Nombre:</b> Guardar información.	
<b>Descripción:</b> Prueba la funcionalidad de guardar los datos capturados.	
<b>Condición de ejecución:</b> <ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #7.</li> </ul>	
<b>Paso de ejecución:</b> <ul style="list-style-type: none"> <li>• Seleccionar la opción “Guardar”.</li> </ul>	

<p><b>Resultados esperados:</b></p> <ul style="list-style-type: none"> <li>Se deben generar los siguientes archivos:             <ol style="list-style-type: none"> <li>Una imagen RGB de la captura clasificada.</li> <li>Un archivo de texto con los 121 puntos de la forma (x; y) del rostro de la persona.</li> <li>Las Unidades de Animación con sus respectivos valores y clasificación, en el formato de archivo de relación de atributos .ARFF.</li> </ol> </li> </ul>
<p><b>Evaluación de la prueba:</b> Satisfactorio.</p>

Tabla 34. Caso de prueba de aceptación #9.

Caso de prueba de aceptación	
<b>Código:</b> HU9p1	<b>Historia de Usuario:</b> 9
<b>Nombre:</b> Cargar modelo de clasificación.	
<b>Descripción:</b> Prueba la funcionalidad de cargar el modelo de clasificación en la aplicación.	
<b>Condición de ejecución:</b>	
<ul style="list-style-type: none"> <li>Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #3.</li> </ul>	
<b>Paso de ejecución:</b>	
<ul style="list-style-type: none"> <li>(Se realiza automáticamente cuando la aplicación es ejecutada por primera vez.)</li> </ul>	
<b>Resultados esperados:</b>	
<ul style="list-style-type: none"> <li>Debe cargarse en la variable especificada el modelo de clasificación.</li> </ul>	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Tabla 35. Caso de prueba de aceptación #10.

Caso de prueba de aceptación	
<b>Código:</b> HU10p1	<b>Historia de Usuario:</b> 10
<b>Nombre:</b> Clasificar la emoción según el modelo utilizado.	
<b>Descripción:</b> Prueba de la funcionalidad clasificar emociones en tiempo real.	
<b>Condición de ejecución:</b>	
<ul style="list-style-type: none"> <li>Debe estar activado el Modo de Clasificación Automática.</li> <li>Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #9.</li> </ul>	
<b>Paso de ejecución:</b>	
<ul style="list-style-type: none"> <li>Activar el Modo de Clasificación Automática.</li> </ul>	
<b>Resultados esperados:</b>	

- El modelo sea capaz de clasificar las instancias correspondientes a las Unidades de Animación capturadas.

**Evaluación de la prueba:** Satisfactorio.

Tabla 36. Caso de prueba de aceptación #11.

Caso de prueba de aceptación	
<b>Código:</b> HU11p1	<b>Historia de Usuario:</b> 11
<b>Nombre:</b> Mostrar resultado.	
<b>Descripción:</b> Prueba la funcionalidad de mostrar el resultado de las clasificaciones de emoción en tiempo real realizadas.	
<b>Condición de ejecución:</b>	
<ul style="list-style-type: none"> <li>• Debe haberse cumplido exitosamente la funcionalidad de la Historia de Usuario #10.</li> </ul>	
<b>Paso de ejecución:</b>	
<ul style="list-style-type: none"> <li>• (Se realiza automáticamente si se cumplen las condiciones definidas.)</li> </ul>	
<b>Resultados esperados:</b>	
<ul style="list-style-type: none"> <li>• Se debe mostrar una clasificación de la emoción que más haya experimentado la persona durante 5 segundos.</li> </ul>	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Se realizaron 11 pruebas de aceptación, una por cada Historia de Usuario, cumpliendo con el procedimiento a seguir. Para un total de 11 pruebas con resultado “Satisfactorio” y ninguna con resultado “Insatisfactorio”.

## CONCLUSIONES DEL CAPÍTULO

- La variante del diseño de obtención de datos de entrenamiento, para esta versión, reflejó resultados más favorables que la elaborada inicialmente.
- El algoritmo de Optimización Secuencial Mínima, basado en las Maquinas de Soporte Vectorial, es el que mejores resultados alcanza para esta versión.

---

## CONCLUSIONES

Con la realización de este trabajo se arribaron a las siguientes conclusiones:

1. Se determinó que el dispositivo Kinect carece de precisión en el reconocimiento de determinados puntos de interés, señalados por Ekman, necesarios para la clasificación de emociones.
2. Las tareas de clasificación de emociones requieren de un proceso iterativo e incremental para optimizar los porcentos de clasificaciones.
3. El SMO es el algoritmo que mejores resultados de clasificación obtuvo para este modelo de datos.



## RECOMENDACIONES

Con el interés de buscar mayor eficiencia en el proceso de clasificación de emociones, se proponen las siguientes recomendaciones:

1. Utilizar la versión 2.0 del Kinect para comprobar si se mejora la precisión en la captura de datos.
2. Incluir a la solución obtenida, la fase de Selección del modelo de clasificación realizada en esta versión en la herramienta WEKA.
3. Integrar la solución desarrollada al software Danzoterapia 2.0.

## REFERENCIAS BIBLIOGRÁFICAS

1. Real Academia Española. Diccionario de la Lengua Española. [En línea] <http://lema.rae.es/drae>.
2. Paul Ekman, Harrieh Oster. *Expresiones faciales de la emoción*. 1979. 527-554.
3. P. Ekman, W.V. Friesen. *The Facial Action Coding System: A Technique for The Measurement of Facial Movement*. San Francisco : San Francisco: Consulting Psychologists Press, 1978.
4. Chóliz, Mariano. Psicología de la emoción: el proceso emocional. [En línea] 2005. [www.uv.es/~cholz](http://www.uv.es/~cholz).
5. FACS - Facial Action Coding System. [En línea] 2002. <http://www.cs.cmu.edu/~face/facs.htm>.
6. "Automated Facial Expression Recognition System" Security Technology. Ryan, Cohn, S. Lucey, Saraigh, P. Lucey, De la Torre, Rossi. s.l. : 43rd Annual 2009 International Carnahan Conference, pp 172-177, 2009.
7. Stocchi, Luca. *3D Facial Expressions Recognition Using The Microsoft Kinect*. Dublin, Ireland : Dublin City University.
8. Cohn, J. F., Ambadar, Z., & Ekman, P. Observer-based measurement of facial expression with the Facial Action Coding System. *The handbook of emotion elicitation and assessment*. N.Y : Oxford University Press Series in Affective Science, 2007.
9. Sergios Theodoridis, Konstantinos Koutroumbas. *Pattern Recognition*. 2003.
10. Benítez, Ing. José C. Procesamiento de Imágenes y Visión Artificial (WEE2) Sesión: 12 Reconocimiento de Patrones. [En línea] [http://es.slideshare.net/jcbp\\_peru/utp-pdiyvasap12-reconocimiento-de-patrones](http://es.slideshare.net/jcbp_peru/utp-pdiyvasap12-reconocimiento-de-patrones).
11. Duda, Richard O., Stork, David G. y E.Hart, Peter. *Pattern Classification*. 2001. 0-471-05669-3..
12. Camila Riveron, Lucía Marroig. *Introducción al Reconocimiento de Patrones. Reconocimiento de cara y desempeño de clasificadores*. 2011.
13. *Top 10 algorithms in data mining*. Wu, Xindong, y otros. 2007. 10115-007-0114-2.
14. Olabe, Xabier Basogain. *Redes Neuronales Artificiales y sus aplicaciones*. Bilbao : E s c u e l a Superior de Ingeniería de Bilbao, EHU. P3BN11.
15. *Application of Microsoft Kinect Sensor for Tracking Construction Workers*. Hubo Cai, Amr Kandil, Makarand Hastak, and Phillip S. Dunston. Calgary, Alberta, Canada : s.n., 2014.
16. *Microsoft Kinect Sensor and Its Effect*. Zeng, Wenjun. 2, s.l. : IEEE Computer Society, Abril de 2012, Vol. 19.
17. Sean Kean, Jonathan C. Hall, Phoenix Perry. *Meet the Kinect*. [ed.] Apress. s.l. : Apress, 2012. 978-1-4302-3888-1.

18. Ladislao Mathe, Darío Samban, Gabriel Gómez. *Estudio del funcionamiento del sensor Kinect y aplicaciones para bioingeniería*. Argentina : IEEE Argencon , 2012.
19. Microsoft Corporation. Microsoft, Face Tracking. [En línea] <https://msdn.microsoft.com/en-us/library/jj130970.aspx>.
20. Ahlberg, Jörgen. Liu, Candide. [En línea] <https://msdn.microsoft.com/en-us/library/jj130970.aspx>.
21. Microsoft Corporation. [En línea] <https://msdn.microsoft.com/en-us/library/jj130970.aspx>.
22. Waikato, Machine Learning Group at the University of Waikato. The University of Waikato. [En línea] <http://www.cs.waikato.ac.nz/ml/weka/>.
23. *El lenguaje de programación C#*. Seco, José Antonio González. 2001.
24. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Letelier, Patricio. 2012.
25. López, José Manuel Molina y Herrero, Jesús García. *Técnicas de análisis de datos*. Madrid : s.n., 2006.
26. Herrera, Francisco, Riquelme, José y Rui, Roberto. *Preprocesamiento de datos*. Madrid : s.n., 2004.
27. Sommerville, Ian. *Ingeniería del software*. Séptima Edición. 2005.
28. *Ingeniería del Software. Un enfoque práctico*. Pressman, Roger S. University of Connecticut : s.n.
29. Waikato, Machine Learning Group at the University of Waikato. Weka, SMO. [En línea] <http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html>.
30. Beck, Kent. *Extreme Programming Explained - 2nd ed.* s.l. : Addison-Wesley Professional, 2012. 0-321-27865-8.
31. Ken Auer, Roy Miller. *Extreme Programming Applied: Playing to Win*. s.l. : Addison-Wesley Professional, 2001. 0201616408.
32. Hastie, T., Tibshirani, J. Friedman. *The Elements of Statistical Learning: Data Mining, Interference and Prediction (Capítulo 12: Support Vector Machines and Flexible Discriminants)*. s.l. : Ed. Springer Series in Statistics, 2001.
33. Pérez., Luis Velázquez. *Reconocimiento Facial Basado en Puntos Característicos de la Cara en entornos no controlados*. Madrid : Universidad Autonoma de Madrid, 2013.
34. Ekman, Paul. *Advances in Neural Information Processing Systems*. 1996.
35. Ekman, Paul, V. Friesen, Wallace y Ancoli, Sonia. Facial Signs of Emotional Experience. *Journal Of Personality and Social Psychology*. San Francisco, San Diego : s.n., 1980, Vol. 39.
36. Paul Ekman, Erika L. Rosenberg. *What the Face Reveals Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*. New York : Oxford University, 1997.

37. Ekman, Paul. *Darwin and Facial Expression: A Century of Research in Review*. s.l. : MALOOR BOOKS, 2006.
38. Ekman, Paul y V. Friesen, Wallace. *Unmasking the Face: A guide to recognizing emotions from facial expressions*. s.l. : MALOR BOOKS, 2003.
39. Paul Ekman, Wallace V. Friesen, Joseph. *Facial Action Coding System*. Estados Unidos : s.n., 2002.
40. Microsoft. Microsoft (hardware). [En línea] <https://dev.windows.com/es-es/kinect/hardware>.
41. Kaufmann, Morgan. *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. 1989. 1-55860-065-5.
42. Figueras, Manuel Salvador. Introducción al Análisis Multivariante. [En línea] 2000. <http://www.5campus.com/leccion/anamul>.
43. Hernández, Roger Gimeno. *Estudio de técnicas de reconocimiento facial*. Barcelona : Departamento de Procesado de Señal y Comunicaciones, Universidad Politecnica de Catalunya, Mayo, 2010.

# ANEXOS

Anexo 1 Archivo .ARFF según el primer diseño del Módulo de Captura de Datos.

```

@RELATION clasific
@ATTRIBUTE valor1 real
@ATTRIBUTE valor2 real
@ATTRIBUTE valor3 real
@ATTRIBUTE valor4 real
@ATTRIBUTE valor5 real
@ATTRIBUTE valor6 real
@ATTRIBUTE valor7 real
@ATTRIBUTE valor8 real
@ATTRIBUTE valor9 real
@ATTRIBUTE valor10 real
@ATTRIBUTE valor11 real
@ATTRIBUTE valor12 real
@ATTRIBUTE valor13 real
@ATTRIBUTE valor14 real
@ATTRIBUTE valor15 real
@ATTRIBUTE valor16 real
@ATTRIBUTE valor17 real
@ATTRIBUTE valor18 real
@ATTRIBUTE valor19 real
@ATTRIBUTE valor20 real
@ATTRIBUTE valor21 real
@ATTRIBUTE valor22 real
@ATTRIBUTE valor23 real
@ATTRIBUTE valor24 real
@ATTRIBUTE valor25 real
@ATTRIBUTE valor26 real
@ATTRIBUTE class {f,t,n}
@data
-0.241679056431278,-0.359221717080161,-0.398002226426893,-0.424568563055339,-0.403876551295568,-
0.365467600829453,-0.233895188973477,-0.22185500924946,-0.1900854849447,-0.218317681503866,-
0.284489466559536,-0.285080944220843,-0.295452179387381,-0.288536869646669,-0.275086831458102,-
0.29827205703528,-0.308191542083808,-0.295681246178573,-0.113286240873522,-0.32601797525137,-
0.735822483928599,-0.366186303597383,-0.234933618919946,-0.382735625434013,-0.743056263369236,-
0.418525863132008,f
-0.159366334264474,-0.290371231493169,-0.334867952077788,-0.308614196976498,-0.337788503069191,-
0.297116309884526,-0.170552886635527,-0.123041174427448,-0.102182391979175,-0.11694931618041,-
0.179808943660333,-0.217105824755151,-0.234212374247174,-0.22327434052373,-0.189175893169174,-
0.20214844608806,-0.207422992431877,-0.196998419252623,-0.207883350359194,-0.305849539999732,-
0.665410326743992,-0.350616282573493,-0.239761026848095,-0.336922498302868,-0.675736491539637,-
0.378192711339981,n

```