

Universidad de las Ciencias Informáticas

Facultad 1



Propuesta de Entorno Colaborativo para el Desarrollo
de Algoritmos Biométricos

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

Autores: Susana Zayas Guerrero

Guillermo Joaquín Martínez Guerra

Tutores: Msc. Yurdik Cervantes Mendoza

Ing. Sonia Guerrero Lambert

Ciudad de la Habana, Junio 2011

“Año 53 de la Revolución Cubana”

DECLARACIÓN DE AUTORÍA

Nosotros, **Guillermo Joaquín Martínez Guerra** y **Susana Zayas Guerrero**, nos declaramos como únicos autores del presente trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que haga uso del mismo de la manera que mejor estime.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Guillermo Joaquín Martínez Guerra

Susana Zayas Guerrero

Msc. Yurdik Cervantes Mendoza

Ing. Sonia Guerrero Lambert

DATOS DE CONTACTO

Tutor: Msc. Yurdik Cervantes Mendoza

Profesor Asistente de la Universidad de las Ciencias Informáticas (UCI).

Graduado de Ingeniero Informático en el Instituto Superior Politécnico José Antonio Echeverría (CUJAE) en el año 2003.

Máster en Informática Aplicada con 5 años de experiencia en Biometría.

E-mail: yurdik@uci.cu

Tutor: Ing. Sonia Guerrero Lambert.

Graduada de Ingeniera en Ciencias Informáticas en la Universidad de las Ciencias Informáticas en el 2008

Profesora del Dpto. de Ingeniería y Gestión de Software de la Facultad 1

Diseñadora principal de bases de datos en el proyecto Nova.

Categoría docente: Instructor.

E-mail: sguerrero@uci.cu

DEDICATORIA

Dedico esta tesis a mi familia y a mi abuela Évora Esther Ginarte Tamayo, aunque ya no se encuentre con nosotros.

Guillermo Joaquín Martínez Guerra.

Mi tesis la dedico con todo mi amor y cariño.

A ti Dios que me diste la oportunidad de vivir y regalarme una familia maravillosa.

A Mimi y mi mamita Nilda, gracias por todo por estar conmigo en todo momento, este trabajo es para ustedes, Mimi por ser tu única hija aquí está lo que tú y mi mamá me brindaron, solamente les estoy devolviendo lo que ustedes me dieron en un principio.

A mis primos Marquitos, Amandita, Ibrainsito y Kent, gracias por estar conmigo y apoyarme siempre, Marquitos estarás constantemente en mis pensamientos, esta Tesis en gran parte es para ti, a ti Amandita solo te doy las gracias por siempre hacerme reír, espero que estés orgullosa de tener una prima ingeniera.

A mis tíos, Negra, Sandy y Niño. Son los mejores del mundo, a pesar de todo lo que hemos pasado somos una familia muy linda, gracias por darme eso.

A mis amigos.

Que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora, seguimos siendo amigos: Dianiseli, Daily, Ismel, Yummy, Anet, Vismar, Wilfredo, Alien y Lachi.

Y a todos aquellos que participaron directa o indirectamente en la elaboración de esta tesis.

¡Gracias a ustedes!

Susana Guerrero Zayas

AGRADECIMIENTOS

Le agradezco infinitamente a mis padres, a mi hermana, a mi abuelo y a mi abuela Esther, por haberme entendido en todo momento, darme su apoyo incondicional y haberme guiado en los momentos más difíciles. Por su cariño y su amor, sin eso no hubiera podido alcanzar las metas que me he trazado en la vida.

A Natacha Lía por haber sido una luz hermosa en mi vida. A Javier por haber sido un hermano para mí todos estos años. A Juan, Natacha, Pepe y toda la familia de mi hermana por tratarme con tanto cariño y darme tan buenos consejos.

A mis abuelos Pedro y Electra y mi tío Ernesto por su cariño y amor.

A Enma por ser mi compañera y amiga y ayudarme en muchos momentos difíciles de mi carrera.

A tía Vivian, abuela Nena, abuelo Kaki, tío Pititi, por quererme como un hijo todo este tiempo.

A todos mis amigos de la universidad, que han sido mi familia todos estos años y nos hemos apoyado unos a otros. Especialmente a Onnier, Leony, Rubén, Diana, Daily, Antonio y el Tocallo, Fino, Uffo, Darién, José Eduardo, El Bichi.

A mi hermanos Eduardo y Manuel, por soportarme todos estos años que llevamos juntos. A Chave, Yadira, Pedro, Julio, por su amistad y cariño.

A mi tutor Yurdik que ha luchado conmigo todo este tiempo para que la tesis salga adelante, sin su ayuda esto no hubiera sido posible. A mi tutora Sonia que también me ayudó y apoyo.

A mi amiga Yayneris por su cariño y sus consejos.

A Susana, mi compañera de tesis; aguantarme a no es fácil, y ella ha sabido llevar esa tarea bastante bien.

Guillermo Joaquín Martínez Guerra.

Le agradezco el poder llegar hasta aquí principalmente a Mimi y mi mamá, Mimi siempre fuiste mamá y papá, nunca, nunca necesité una figura paterna por tu dedicación hacia mi persona, nunca había tenido la oportunidad de agradecértelo, este título es mi humilde mi forma de hacerlo. Tú y mi mamá son el aire que respiro, sin ustedes no soy nadie, mi razón de ser y mi orgullo, gracias por estar en mi pasado, mi presente y mi futuro, no me equivoco si digo que son las mejores mamás del mundo, gracias por todo su esfuerzo, apoyo y por la confianza que depositaron en mí. Gracias porque siempre, aunque lejos, han estado a mi lado. Las quiero mucho.

A mis amistades, ustedes fueron sin duda alguna mi familia estos 5 años en la UCI, gracias por aguantarme y compartir tan buenos momentos conmigo, a ustedes: Dianiseli, Daily, Lienys, Popi, Ariannis, Ismel, Gustavo, Yumy, Anet, Mairelis y Tahimí.

Ismel, mi gordito lindo, te debo mucho, nunca voy a poder pagarte el haberme escogido como amiga, espero que nuestra amistad dure por siempre, a pesar de la lejanía que ahora nos pueda distanciar, nunca dudes de que siempre vas a tener un lugar en mi corazón, te quiero mucho.

A mi entrenadora Yunelsis, gracias por meterme el bichito del futbol, y por convertirte en una amiga muy querida.

A mi tutor Yurdik, sin tu ayuda jamás hubiese logrado esta Tesis, muchísimas gracias.

Y a mis compañeras de apartamento.

Gracias....

Susana Zayas Guerrero.

FRASE

“El hombre ordinario sólo se cuida de pasar el tiempo; el hombre de talento, de emplearlo”



Arthur Schopenhauer (1788 – 1860).

Filósofo alemán.

RESUMEN

Gracias al interés creciente de la Revolución por lograr un desarrollo en todas las ramas de la sociedad, incluyendo la Informática, en Cuba se han creado centros dedicados a la investigación y creación de productos informáticos de todo tipo. Uno de estos centros es el Centro de Identificación y Seguridad Digital ubicado en la Universidad de las Ciencias Informáticas, en la que se desarrollan actualmente algoritmos biométricos mediante un proceso complejo y demorado, debido esto principalmente a la falta de integración de herramientas especializadas y la ausencia de un proceso de desarrollo de algoritmos biométricos bien definido.

La presente tesis tiene como objetivo realizar una propuesta de Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos, donde se definen un conjunto de tecnologías y un proceso de desarrollo de algoritmos biométricos que le dan soporte. Para lograr dicho objetivo se realiza un estudio de los entornos colaborativos existentes en el mundo en diversas ramas de la sociedad, para profundizar en sus características principales así como en las herramientas que fueron utilizadas en ellos, principalmente las que propician la colaboración. Los sistemas biométricos así como los algoritmos que soportan el funcionamiento de estos, también son objetos de estudio, junto a las principales prácticas de diseño y las herramientas más utilizadas en el desarrollo de algoritmos biométricos.

Por último se describe el proceso realizado para la validación de la propuesta mediante la técnica de Criterio de Expertos, lo que permite evaluar la calidad de la misma.

Palabras claves: Colaboración, Entorno Colaborativo, Algoritmo Biométrico, Proceso.

ÍNDICE

INTRODUCCIÓN.....	1
CAPITULO 1: FUNDAMENTACIÓN TEÓRICA.....	1
1.1 ¿QUÉ ES LA BIOMETRÍA?	1
1.1.1 Tipos de Biometría.....	1
1.2 SISTEMA BIOMÉTRICO	1
1.1.2 Estilos arquitectónicos (Arquitecturas).....	2
1.3 ALGORITMO BIOMÉTRICO	3
1.3.1 Comparación de los métodos.....	6
1.3.2 Algoritmos Biométricos en Cuba.	8
1.3.3 Algoritmos Biométricos en la Universidad de las Ciencias Informáticas (UCI).	8
1.3.4 Herramientas para el diseño de algoritmos.....	8
1.3.5 Herramientas de programación visual.	10
1.3.6 Herramientas gráficas de diagramas en bloques (HGDB).....	11
1.4 DESARROLLO COLABORATIVO.	12
1.4.1 Proceso de desarrollo colaborativo.	12
1.4.2 Entorno Colaborativo.....	13
1.5 HERRAMIENTAS PARA LA GENERACIÓN DE REPORTES.	23
1.5.1 Report Manager.	24
1.5.2 Pentaho Reporting.	24
1.5.3 iReports.....	25
1.5.4 Comparación entre los diseñadores de reportes estudiados.....	27
1.6 HERRAMIENTAS PARA EL CONTROL DE VERSIONES.	27
1.6.1 CVS (Concurrent Versions System).....	28
1.6.2 Subversion.....	28
1.6.3 Git.....	29
1.6.4 Mercurial.	30
1.7 HERRAMIENTAS DE GESTIÓN DE BASES DE DATOS.	30
1.7.1 PostgreSQL.....	30
1.7.2 SQLite.	31
1.7.3 MySQL.	31
1.8 HERRAMIENTAS DE ENTORNO DE DESARROLLO INTEGRADO.	31
1.8.1 Matlab.....	32
1.8.2 Octave.	33
1.9 PLATAFORMA LEGAL. LICENCIAS	33
1.10 CONCLUSIONES DEL CAPÍTULO	34
2 CAPITULO 2: PROPUESTA DE SOLUCION.....	35
2.1 CICLO DE DESARROLLO.	36
2.1.1 Plataforma de Integración.....	36
2.1.2 Fase de Diseño	37
2.1.3 Fase de Implementación.....	40
2.1.4 Fase de Pruebas.	41
2.1.5 Herramientas colaborativas.....	47
2.2 CONCLUSIONES DEL CAPÍTULO.....	48
3 CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA POR VALORACIÓN DE EXPERTOS.....	49
3.1 VALIDACIÓN DE LA PROPUESTA.	49
3.1.1 Panel de Expertos.	49
3.1.2 Selección del panel de expertos.....	49

3.1.3	<i>Elaboración de la encuesta.</i>	50
3.1.4	<i>Guía para la evaluación de la propuesta.</i>	50
3.2	CONCLUSIONES DEL CAPÍTULO	52
	CONCLUSIONES.	53
	RECOMENDACIONES.	54
	REFERENCIA BIBLIOGRÁFICA.	55
	BIBLIOGRAFÍA.	58
	GLOSARIO DE TÉRMINOS.	60
	ANEXOS.	63

ÍNDICE DE TABLAS

TABLA 1: LICENCIAS DE SOFTWARE LIBRE	34
TABLA 2: FILTROS PROPUESTOS	39
TABLA 3: TABLA ESTADÍSTICA COMPARATIVA DONDE P017,....., P072, SON ALGORITMOS. (SEGÚN EL FVC 2006). ...	45
TABLA 4: RESUMEN DE LA EVALUACIÓN EMITIDA POR LOS EXPERTOS.....	51

INDICE DE FIGURAS

FIGURA 1: PROCESO DE INSCRIPCIÓN	1
FIGURA 2: PROCESO DE VERIFICACIÓN	1
FIGURA 3: PROCESO DE IDENTIFICACIÓN.....	1
FIGURA 4: TASA DE ERROR IGUAL.	4
FIGURA 5: MUESTRA EL ZEROFNMR Y EL ZEROFMR	5
FIGURA 6: EJEMPLO DE UNA VISTA DE HGDB.....	11
FIGURA 7: PANTALLA DE LA APLICACIÓN REPORT MANAGER DESIGNER	24
FIGURA 8: VISTA DEL REPORT DESIGNER.....	25
FIGURA 9: FUNCIONAMIENTO DE IREPORT	26
FIGURA 10: MUESTRA DEL CÓDIGO PARA UN JRXML.....	26
FIGURA 11: VISTA DEL IREPORT.....	27
FIGURA 12: DESCRIPCIÓN GENERAL DEL ENTORNO.	36
FIGURA 13: SE VE EL COMPORTAMIENTO DE LOS ALGORITMOS: ERR, ..., FRR. (SEGÚN EL FVC 2006).....	46

INTRODUCCIÓN

Desde la prehistoria el hombre como ser pensante necesitó identificarse como ser único y con características propias. Identificación, vigilancia, control, no son conceptos del mundo moderno, sino que caminan de la mano de la historia del hombre. Tanto es así que se tienen datos sobre pinturas en cuevas rodeadas de huellas de manos usadas aparentemente como “firma” que datan del año 31.000 a.C. En 500 a.c se muestran huellas digitales usadas en transacciones de negocio en Babilonia, marcadas en arcilla. También hay constancia del uso de huellas digitales en China, incluso por padres para diferenciar a sus hijos.[1]

En el siglo XIX comienzan las investigaciones científicas acerca de la biométrica con el fin de buscar un sistema de identificación de personas con fines policiales. Con la llegada de los computadores se revoluciona el modo en que la información era almacenada desde los años '70, eso significó que las huellas dactilares, como única técnica biométrica utilizada en esa época, ahora podían ser digitalizadas y almacenadas de un modo que hace mucho más fácil acceder a ellas y compararlas. Esto es más eficiente comparado con el viejo sistema manual que se usaba en años anteriores para la identificación y verificación de las personas.

En 1936 se sugiere el uso del iris como rasgo biométrico. Ocurre lo mismo en el año 1960 con el empleo semiautomático de rasgos faciales. En 1965 comienza el uso del habla también como rasgo biométrico, y en los años 70 se inician los pasos para la automatización de los procesos y el nacimiento de estándares y patentes.

Hoy, sistemas automáticos que escanean y digitalizan los datos necesarios del ente a analizar, han llevado una técnica mucho más rigurosa, donde se pueden encontrar todo tipo de dispositivos biométricos para controlar los accesos a sistemas informáticos, garantizar la seguridad en transacciones bancarias o simplemente acceder a cierto local con un alto grado de confidencialidad. Ante la necesidad de sistemas cada vez más seguros el hombre ha recurrido a la Biometría aplicada a la verificación de la identidad de un individuo de forma automática, empleando sus características biológicas, psicológicas y de conducta. Esta identificación, que es la única que permite una autenticación individual y exacta, utiliza ciertos patrones fisiológicos, digitalizados y almacenados. [2]

Todos estos sistemas presentan un procedimiento bastante complejo, con características y elementos de gran importancia, entre los cuales se encuentran los algoritmos biométricos que lo respaldan (extracción y cotejo). De la calidad de estos dependen en gran medida los resultados finales de dichos sistemas, ya que representa un componente principal en toda la gama de operaciones que realizan, entre las que se encuentra captura y comparación de los datos, entre

otros. Los algoritmos para la biometría se encuentran en continuo desarrollo y perfeccionamiento. En la actualidad, estos se siguen evaluando por medio de métricas que describen la precisión y el desempeño computacional.

El Departamento de Biometría del Centro de Identificación y Seguridad Digital (CISED), desarrolla actualmente algoritmos biométricos mediante un proceso complejo y demorado dado que los desarrolladores no disponen de herramientas especializadas que los ayuden en cada una de las fases de su desarrollo. El único algoritmo biométrico desarrollado hasta el momento en dicho departamento se obtuvo a partir de un exhaustivo proceso de desarrollo que involucra gran cantidad de esfuerzo y tiempo.

Parte de este tiempo se invierte midiendo la precisión del algoritmo con herramientas independientes que carecen de la calidad necesaria. Estas herramientas son confeccionadas por los desarrolladores, que utilizan una base de datos de pruebas de referencia específica y no tienen la flexibilidad de incorporar otras bases de datos. Estas herramientas no están integradas en una interfaz gráfica usable por lo que requieren de mayor intervención especializada de los usuarios.

Las herramientas, procedimientos y artefactos no están formalizados en un proceso que permita realizar de manera óptima la colaboración entre un grupo de especialistas en el análisis de la calidad de los algoritmos y los factores que influyen sobre esta. Los procesos empíricos actuales no permiten el análisis inmediato y transparente por no utilizar un repositorio de versiones ni herramientas que permitan la comunicación entre los especialistas.

Tampoco se guardan tablas de comparaciones estadísticas de la precisión de los algoritmos para un posterior análisis de su evolución. En el proceso actual de implementación del algoritmo no se realiza una comparación automática de los resultados obtenidos con una BD de pruebas de nivel internacional, ni se permiten añadir nuevas, además no se guardan versiones previas de los artefactos obtenidos (algoritmos) ni la información sobre la precisión de ellos para poder sugerir nuevos cambios. A la par de esto, los algoritmos no pueden ser etiquetados con marcadores semánticos que señalen las características distintivas de cada uno para el futuro análisis de estos combinados, o sea, que no tenemos un resultado de un sistema que realice análisis automático de la composición del algoritmo y propuestas de las técnicas que han sido más efectivas para utilizar.

Atendiendo lo anteriormente planteado se llega a la siguiente **situación problemática**: El desarrollo de algoritmos biométricos en el Dpto. de Biometría del CISED es demorado, complejo y adolece de herramientas colaborativas. Esto atenta contra la terminación en tiempo de los proyectos, se realiza trabajo redundante y existe poca información para el análisis estadístico sobre el resultado de las

pruebas. Situación que da paso al **problema científico**: ¿Cómo simplificar y acelerar el proceso de desarrollo de algoritmos biométricos del Departamento de Biometría del CISED?

Dicha situación amerita que se realice el presente trabajo, para alcanzar una alternativa de solución. El **objeto de estudio** queda enmarcado en: Los entornos de desarrollo colaborativo. El **campo de acción** queda delimitado en: Los entornos de desarrollo colaborativo de algoritmos biométricos en el CISED. Se persigue como **objetivo general**, elaborar una propuesta de entorno colaborativo que permita el desarrollo de algoritmos biométricos.

Objetivos específicos:

- ✓ Diseñar un proceso para el desarrollo colaborativo de algoritmos biométricos (roles, actividades, interacciones, descripción de procedimientos).
- ✓ Definir el conjunto de herramientas que permitan soportar el proceso propuesto e integrarlas en un entorno de desarrollo colaborativo. Elaborar los requerimientos funcionales de las herramientas que no estén disponibles.
- ✓ Validar la propuesta de solución.

Se plantea como **idea a defender**, que la definición de un proceso, los métodos y herramientas colaborativas para el desarrollo de algoritmos biométricos en el CISED, permitirá simplificar y acelerar el desarrollo de algoritmos biométricos en dicha entidad y mejorar la calidad de estos.

Métodos de investigación

Teóricos

- ✓ Analíticos – sintéticos: se utiliza en el proceso de análisis de la bibliografía empleada, realizando una síntesis y arribando a conclusiones que contribuyen a la propuesta final.

Empíricos

- ✓ Entrevista: se utiliza en la investigación para precisar el problema a resolver y para conocer los principales motivos que impulsan el desarrollo del presente trabajo de diploma.
- ✓ Encuesta: se emplea para dar validación a la propuesta del Entorno Colaborativo para el desarrollo de Algoritmos Biométricos.

El trabajo está estructurado de la siguiente manera:

En el **Capítulo 1** se hace referencia a los conceptos básicos de Biometría, los algoritmos biométricos y su proceso de desarrollo. Además se analiza la posibilidad de que estos procesos de desarrollo

puedan ser colaborativos y la aplicabilidad actual de este tipo de procesos en la interacción del hombre con la sociedad. Se estudian y describen varios entornos colaborativos actuales.

En el **Capítulo 2** se realiza la propuesta de solución, donde se define un proceso para el desarrollo de algoritmos biométricos. Se especifican varios aspectos esenciales dentro de dicho proceso, como por ejemplo las fases con las que contará y sus respectivas actividades, los roles que intervienen en cada fase o actividad, la comunicación entre los distintos roles, así como las características del ciclo de desarrollo y los artefactos generados en las distintas fases. Se propone un conjunto de herramientas las cuales serán utilizadas en las distintas fases del proceso de desarrollo colaborativo. Se definen los métodos mediante los cuales los involucrados en los distintos roles utilizarán las herramientas propuestas para ejecutar las distintas actividades del proceso propuesto. Por último se explica cómo interaccionan todas las herramientas propuestas.

En el **Capítulo 3** se realiza la validación de la propuesta por Criterio de Expertos.

CAPITULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se hace referencia a los conceptos básicos de Biometría, los algoritmos biométricos y su proceso de diseño. Además se analiza la posibilidad de que estos procesos de desarrollo puedan ser colaborativos y se analiza la aplicabilidad actual de este tipo de procesos en la interacción del hombre con la sociedad. Se estudian y describen varios entornos colaborativos actuales relacionados con el tema de investigación.

1.1 ¿Qué es la Biometría?

La biometría es una rama de la biología que estudia y mide los datos de los seres vivos. El término "biometría" deriva de la palabra griega "bios" (vida) y "metros" (medida) y en su significado corriente en informática, sirve para indicar la identificación automática o la verificación de la identidad de un sujeto, sobre la base de sus características físicas o de su comportamiento. [3]

Biometría Informática:[4]

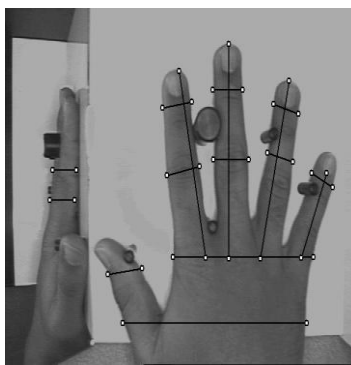
- ✓ La "Biometría Informática" es la aplicación de técnicas biométricas a la autenticación e identificación automática de personas en sistemas de seguridad informática.
- ✓ Las técnicas biométricas se basan en medir al usuario, directa o indirectamente, para reconocerlo automáticamente, aplicando técnicas estadísticas y de inteligencia artificial (lógica borrosa, redes neuronales, entre otras).

1.1.1 Tipos de Biometría

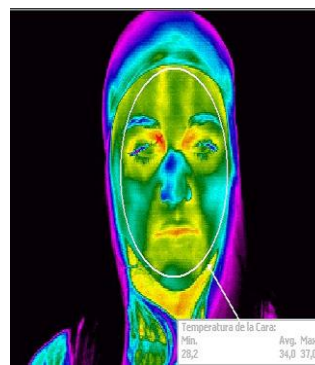
- **Biometría Estática:** Es la que mide la anatomía del usuario. Comprende entre otras: [5, 6]



a) Huellas Digitales



b) Geometría de la mano



c) Termografía

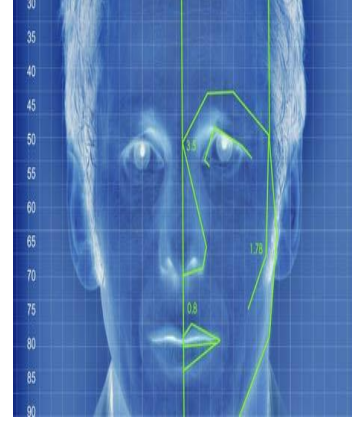


d) Análisis del iris



Análisis de retina.

e) Análisis de retina



f) Reconocimiento Facial

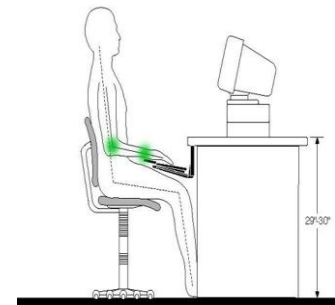
- **Biometría Dinámica:** Es la que mide el comportamiento del usuario. Comprende entre otras:



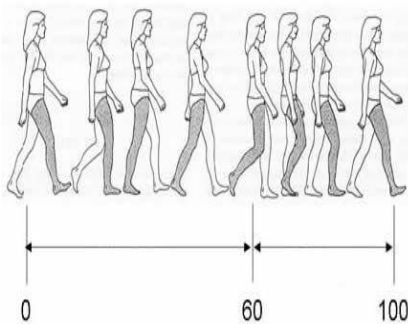
a) Patrón de Voz



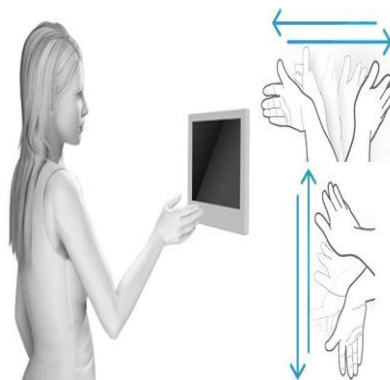
b) Firma manuscrita



c) Dinámica de tecleo



d) Cadencia del paso



e) Análisis gestual

1.2 Sistema Biométrico

Según, (Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar) 2009 [7] , un sistema biométrico es un sistema de dispositivos y componentes de software cuya función fundamental es determinar cómo va a ser reconocido un individuo. A continuación se muestran tres procesos biométricos fundamentales dependiendo del contexto de aplicación:

1. **Proceso de inscripción**¹: El sistema se carga con las muestras biométricas de los individuos que serán identificados por este. Las muestras sirven de referencia para futuras comparaciones biométricas.

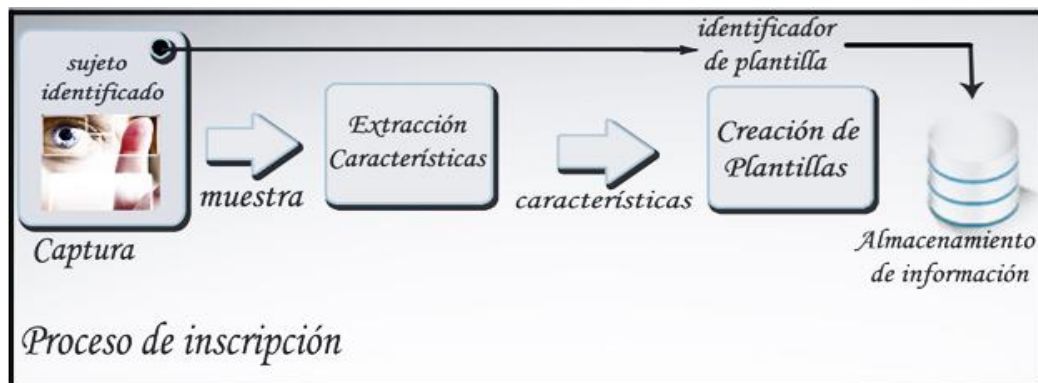


Figura 1: Proceso de Inscripción

2. **Proceso de verificación**: El sistema comprueba la identidad de un individuo (si es quien dice ser) utilizando su muestra biométrica y comparándola con sus características biométricas previamente almacenadas en la fase de inscripción.

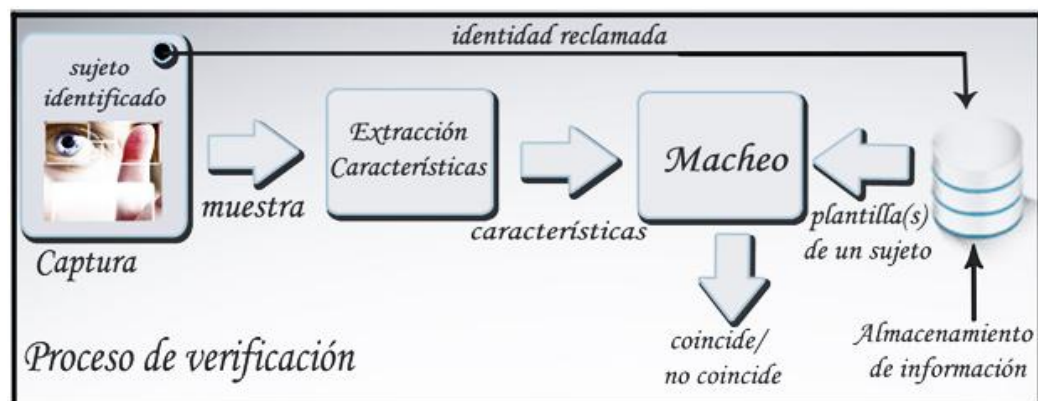


Figura 2: Proceso de Verificación

3. **Proceso de identificación**: El sistema reconoce a un individuo mediante una búsqueda comparativa de la muestra en la toda la base de datos biométrica del sistema. En este proceso se devuelven las **identidades** de los individuos cuyas

¹ también llamado proceso de enrolamiento

características biométricas presentan un grado alto de similitud con las del individuo buscado.

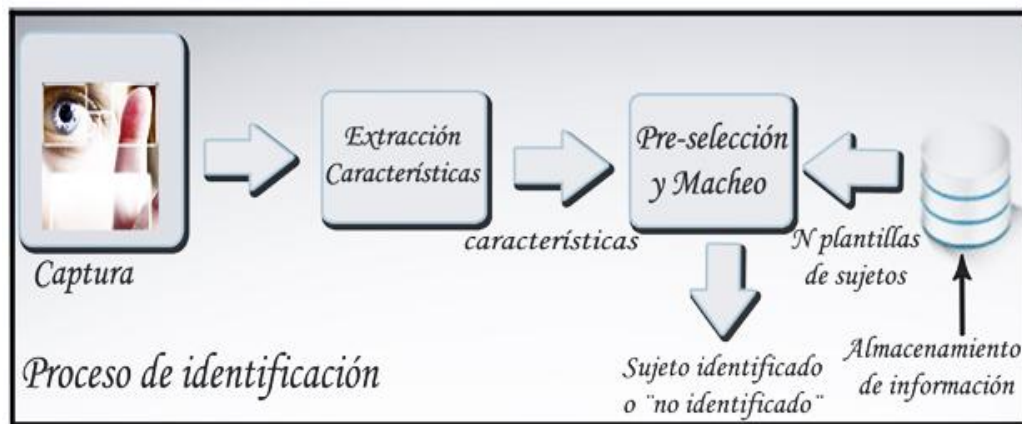


Figura 3: Proceso de Identificación

En estos procesos, el sistema biométrico utiliza sus componentes de:

1. **Captura:** se comunican con el sensor biométrico y obtienen una representación en formato digital de la muestra biométrica obtenida del individuo.
2. **Extracción de características:** Procesan la muestra biométrica y busca patrones biométricos invariables que puedan ser identificados en sucesivas muestras del usuario (características biométricas).
3. **Creación de plantillas biométricas:** Los conjuntos de características biométricas son agrupadas en una representación matemática llamadas plantillas biométricas.
4. **Coincidencia (o cotejo):** Las comparaciones biométricas son especiales, no se comparan las características para verificar que sean iguales (en realidad nunca son iguales) sino para verificar con algoritmos un grado de similitud.
5. **Preselección:** Es un proceso de filtrado que se aplica a la base de datos del sistema con criterios de bajo costo de procesamiento que discriminan gran cantidad de plantillas biométricas para la fase de cotejo. Este componente es usado mayormente en sistemas de gran cantidad de muestras.
6. **Decisión:** El resultado de la operación de cotejo es un grado de similitud (score) que debe ser interpretado por el sistema para decidir si las muestras comparadas pertenecen a la misma persona.
7. **Almacenamiento de datos:** Almacena las plantillas y otra información demográfica del usuario.

1.1.2 Estilos arquitectónicos (Arquitecturas).

La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema. Establece los fundamentos para que analistas, diseñadores, programadores, y otros, trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. [8]

Algunas de las arquitecturas más usadas son:

Cliente-servidor. Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD). Por otro lado los clientes suelen ser estaciones de trabajo que solicitan varios servicios al servidor. Ambas partes deben estar conectadas entre sí mediante una red. [9]

Tuberías y Filtros. Se descompone el sistema en módulos funcionales. Los datos llegan a un filtro, se transforman y son pasados a través de tubos al siguiente filtro.[10]

Arquitectura de tres niveles. Generalización de la arquitectura cliente-servidor donde la carga se divide en tres partes con un reparto claro de funciones: una capa para la presentación, otra para el cálculo y otra para el almacenamiento. Una capa solamente tiene relación con la siguiente. [11]

Arquitectura Orientada a Servicios (SOA). Establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. [12]

Modelo Vista Controlador (MVC) Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web. [13]

Como se puede apreciar en la Figura 2, en el proceso de verificación se encuentra el componente que procesa la imagen (plantilla) y extrae sus características. Dicho componente responde a un proceso donde a la imagen se le van aplicando operaciones sucesivas por lo que suele utilizarse el estilo arquitectónico de tuberías y filtros al implementarlo. Las imágenes procesadas son sometidas a una serie de operaciones (filtros) secuenciales de las cuales se van obteniendo diferentes resultados, que contribuyen todos a obtener uno de carácter final. Dicha arquitectura consta de un número de componentes denominados filtros conectados entre sí por tuberías que transmiten datos desde un

componente al siguiente. Cada paso de procesamiento se encapsula en un filtro. Los datos se pasan usando los tubos entre filtros adyacentes. Cada filtro trabaja independiente de los componentes que se encuentran situados antes o después de él. Se diseña de tal manera que se espera un conjunto de datos en un determinado formato y se obtiene como resultado otros datos de salida con un formato específico. [7]

1.3 Algoritmo Biométrico

Según Cormen, T.H, [14] un algoritmo es un **procedimiento computacional** bien definido, que toma un conjunto de valores de **entrada** y produce un conjunto de valores de **salida**, para resolver un **problema computacional** bien especificado. En el caso de la biometría, los problemas computacionales a resolver responden a las necesidades de algunos de los procesos anteriormente expuestos. Los algoritmos fundamentales, utilizados en los componentes de un sistema biométrico son:

1. **Algoritmos de extracción de características:** Basados en el procesamiento sucesivo de imágenes con representación de la muestra biométrica, su objetivo es obtener los rasgos matemáticos distintivos del individuo. Se utilizan en los componentes de **captura** (para el chequeo de calidad de la muestra capturada²) y en el componente de **extracción de características**. [15]
2. **Algoritmos de cotejo (o de coincidencia):** Determinan el grado de similitud entre las características de dos muestras para verificar si pertenecen o no al mismo individuo. Se realizan operaciones matemáticas y algorítmicas para obtener un valor que representa el grado de similitud (score). Un algoritmo tiene un número finito de pasos y se suele utilizar por el motor biométrico para calcular si una muestra biométrica y la plantilla son equivalentes, es decir que pertenezcan a la misma persona. [16]

En los sistemas biométricos, la **precisión** y el **rendimiento** son dos factores fundamentales a tener en cuenta en su evaluación, frecuentemente estas dos características dependen del buen diseño de los algoritmos utilizados en los procesos previamente enumerados.

En un mundo tan competitivo como el actual, en donde las tecnologías de la informática avanzan a pasos agigantados, se hace cada vez más necesario una mayor calidad en cada producto que sale al mercado. El caso de los algoritmos biométricos y los sistemas biométricos en general, es un ejemplo fehaciente de lo antes dicho. Muchas instituciones llevan décadas estudiando este tema, con el objetivo de crear algoritmos biométricos óptimos. Han definido métricas o índices mediante los cuales se puede medir tanto la precisión, como el rendimiento de dichos algoritmos. Algoritmos que

² “Dado que la calidad biométrica tiene estrecha relación con la cantidad de características detectadas en la muestra de manera efectiva”.

competitivamente pueden poner en la punta la industria de una empresa. A continuación se hace mención de ellos para un mayor entendimiento:[17]

- ✓ **Tasa de Falsos Aceptados (FAR) (FMR)³:** Es la probabilidad de que un individuo no autorizado sea autenticado. A nivel de fabricantes la mayoría tienen esta tasa entre el 0.0001% y el 0.1%. En la práctica el FAR puede ser estimado de la siguiente forma:

$$\text{FAR} = \frac{\text{Número de Falsos Aceptados}}{\text{Número de intentos del impostor}}$$

- ✓ **Tasa de Falsos Rechazados (FRR) (FNMR)⁴:** Es la probabilidad de que un individuo autorizado sea de manera no apropiada rechazado. Comercialmente su valor varía entre el 0.00066% y el 1%. En la práctica el FRR puede ser estimado de la siguiente forma:

$$\text{FRR} = \frac{\text{Número de Falsos Rechazados}}{\text{Número de intentos del cliente}}$$

- ✓ **Tasa de Error Igual (EER):** El punto de intercepción entre la tasa de falsa aceptación y la tasa de falso rechazo. Algunas veces se llama tasa de error cruzada (CER). Es una estadística que muestra la actuación del sistema biométrico, típicamente cuando opera en la tarea de verificación. En general entre más bajo sea el valor de la tasa de error igual, más alto es la precisión del sistema biométrico.

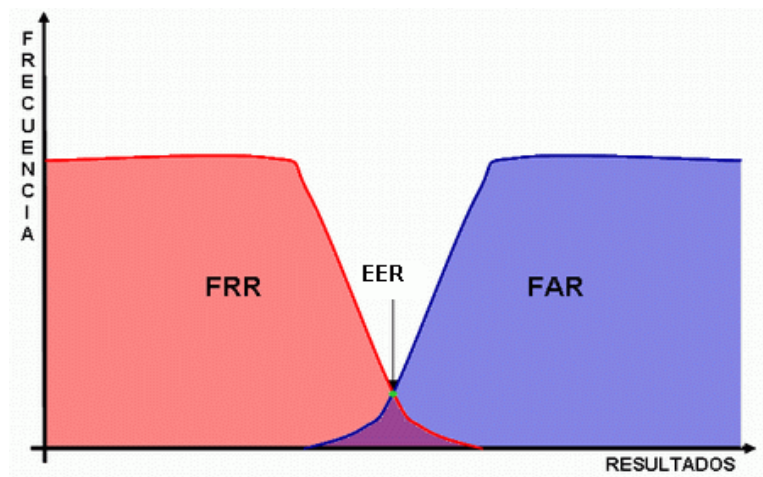


Figura 4: Tasa de Error Igual.

³ False Match Rate

⁴ False Non Match Rate

- ✓ **ZeroFMR (CeroFAR):** Es definido como la menor FRR cuando el FAR tiende a 0 (0%).
- ✓ **ZeroFNMR (CeroFRR):** Es definido como la menor FAR cuando el FRR tiende a 0 (0%).

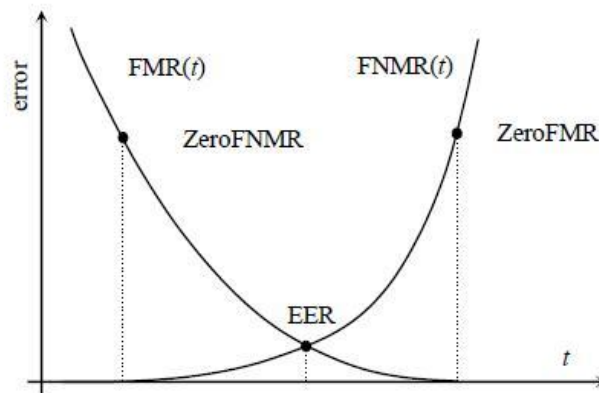


Figura 5: Muestra el ZeroFNMR y el ZeroFMR

- ✓ **FMR100 (FAR100):** Está definido como el menor FNMR (FRR) cuando el FMR (FAR) $\leq 1\%$. (Ver Figura 13).
- ✓ **FMR1000(FAR1000):** Está definido como el menor FNMR (FRR) cuando el FMR (FAR) $\leq 0,1\%$. (Ver Figura 13).
- ✓ **REJENROLL:** Número de huellas dactilares rechazada durante la extracción de características (minucias).
- ✓ **REJMATCH:** Número de huellas dactilares rechazadas durante la comparación.

Los anteriormente mencionados son los principales factores en cuanto a **precisión** por los que se puede evaluar un algoritmo. Existen otros que miden el **rendimiento** y que no dejan de ser importantes, estos son: [18]

- ✓ **(AET) (Average enrollment time)** Promedio de tiempo de la extracción de características (minucias).
- ✓ **(AMT) (Average matches time)** Promedio de tiempo de comparación.
- ✓ **(ATS) (Average template size)** Tamaño promedio de la plantilla.
- ✓ **(MTS) (Maximum template size)** Máximo tamaño de plantilla.
- ✓ **(MEM) (Maximum enrollment memory)** Máximo espacio que ocupa en memoria al hacer la extracción de minucias.

- ✓ **(MMM) (Maximum matches memory)** Máximo espacio que ocupa en memoria al hacer la comparación.

1.3.1 Comparación de los métodos

Desde 1970 se han realizado una gran variedad de algoritmos biométricos, seguidamente se presenta una breve explicación de cómo se desempeñan en su proceso de comparación biométrica.

Algoritmos para verificar huellas dactilares [19]

La huella dactilar ha sido siempre el rasgo biométrico utilizado por la humanidad, durante siglos, para la identificación de las personas. Es un rasgo particular de cada individuo, cuyo origen tiene lugar durante la etapa fetal y permanece inmutable a lo largo de toda la vida. La huella dactilar permite además, discriminar perfectamente a los individuos y su grado de aceptabilidad es relativamente alto. No obstante se precisa de cierta cooperación por parte del individuo, para que la imagen adquirida de la huella tenga la suficiente calidad, como para permitir el empleo de algoritmos de reconocimiento sencillos. En muchas aplicaciones, las condiciones de adquisición no son lo suficientemente favorables y, por tanto, la mala calidad de las huellas adquiridas obliga al empleo de algoritmos complejos, tanto en la etapa de extracción de características, como en la etapa de reconocimiento de patrones.

La forma general de la huella digital se utiliza generalmente para pre-procesar las imágenes, y reducir la búsqueda en grandes bases de datos. La mayoría de los algoritmos están utilizando las minucias y los puntos específicos, como las crestas que termina y las bifurcaciones. Sólo la posición y dirección de estas características se almacenan de la huella para la comparación posterior.

Algunos algoritmos cuentan el número de crestas entre los puntos en particular, por lo general los pequeños detalles, en lugar de las distancias calculadas a partir de la posición. Algoritmos de coincidencia de patrones están utilizando la forma general de las crestas. La huella digital se divide en pequeños sectores, y la dirección del canto, la fase y el tono se extraen y se almacenan.

Algoritmo de reconocimiento de rostro. [20]

En el rostro existe abundante información para reconocer el estado mental y humor de un individuo. En todas las caras existen al menos dos ojos, dos cejas, la nariz, la boca y la barbilla. La distancia entre los ojos, la sombra de las cejas, la nariz y la barbilla son diferentes para cada individuo. Al igual que cambia el tamaño, los ángulos o la expresión de la cara, varían también las imágenes de la cara. La relación de la posición de las distintas partes de la cara, así como sus sombras y tamaños, ha contribuido a la clasificación de los rostros y a la posibilidad de reconocer a una persona.[21]

El reconocimiento de la cara bidimensional ha experimentado algunos obstáculos que el reconocimiento de la cara tridimensional tradicionalmente elimina parcial o totalmente: la iluminación consistente de una cara y las sombras correspondientes; orientación común o pose de una cara; y

variación de las expresiones faciales. Debido al juego más rico de pistas geométricas tridimensionales, incluso la información del rango (por ejemplo, profundidad), el descubrimiento de la cara puede simplificarse. La habilidad inherente de sistemas tridimensionales de reconocimiento facial compensa parcial o totalmente la pose, iluminación, y la expresión puede necesitarse en guiones en que el ambiente de la captura no se controla, como en un cajero automático. La mayoría de cajeros automáticos no ha controlado la iluminación y ha requerido una pose o expresión específica. [22]

Cada registro debe pertenecer a un solo individuo y debe contener una o más imágenes de la cara humana. Dependiendo del tipo de imagen facial, una representación 3D de la cara puede incluir adicionalmente una imagen

Existe un algoritmo llamado **VERILOOK**, el cual está basado en la tecnología de reconocimiento de rostros **VeriLook PC** y es usada por desarrolladores e integradores de sistemas biométricos. Este garantiza un rápido desarrollo de aplicaciones usando funciones de la librería **VeriLook**, la cual asegura un alto nivel de confiabilidad en la identificación de rostros.

VeriLook puede ser fácilmente integrado en los sistemas de seguridad del cliente y las funciones de SDK⁵ pueden ser utilizadas con la mayoría de las cámaras fotográficas (incluyendo Webcams) con cualquier base de datos y con cualquier interfaz de usuario.

VeriLook SDK incluye una librería controladora de cámaras para Microsoft Windows, el cual optimiza la captura simultánea desde múltiples cámaras.

Algoritmo de análisis de Iris. [23]

La tecnología de reconocimiento del iris mira las características únicas del iris. Mientras la mayoría de biométricos tiene 13 a 60 características distintas, se dice que el iris tiene 266 puntos únicos. Se cree que Cada ojo es único y permanece estable con el tiempo y en los ambientes (el ej., tiempo, el clima, las diferencias profesionales). [22]

VERIEYE: El iris del ojo humano es tan único como las huellas dactilares. Cada uno de nuestros ojos posee un iris diferente con un patrón completamente único y a menos que sufra heridas, permanece sin cambios el resto de nuestras vidas. El iris se captura a distancia, usualmente en la región espectral casi infrarroja en la que incluso ojos claros y oscuros muestran una rica textura.

Esta complejidad y abundancia de texturas, combinada con la variedad de personas, hace posible utilizar el reconocimiento de iris en ciertos escenarios de identificación biométrica. Un sistema de identificación de iris puede ser diseñado utilizando:

⁵ *Software Development Kit.*

- ✓ Una PC común, con suficientes recursos para ejecutar rápidos reconocimientos y comparaciones.
- ✓ Un escáner o cámara para iris, actualmente VeriEye es compatible con los escáneres Cross Match I Scan 2 y Retica Mobile-Eyes.
- ✓ Software, que pueda ser rápidamente desarrollado con un SDK que posea todas las tecnologías necesarias, algoritmos e interfaces.

1.3.2 Algoritmos Biométricos en Cuba.

En Cuba hay pocos estudios publicados sobre la creación de algoritmos biométricos, pero esto no significa que no se estén realizando esfuerzos para lograr avanzar en dicho campo. El Centro de Aplicaciones de Tecnología de Avanzada (CENATAV) ha logrado algunos avances en el tema, como por ejemplo en el reconocimiento de patrones, específicamente en las formas de representación, algoritmos de comparación de impresiones palmares y algoritmos de reconocimiento de rostro usando tecnología GPU, entre otros.

1.3.3 Algoritmos Biométricos en la Universidad de las Ciencias Informáticas (UCI).

En la Universidad de Ciencias Informáticas hasta la fecha hay un algoritmo en desarrollo pero avanza muy lentamente. Cuando uno de nuestros centros desarrolladores de técnicas biométricas necesita uno de estos algoritmos, se recurre a componentes propietarios del mercado que implementan dichos algoritmos.

Por lo antes expuesto se puede llegar a la conclusión de que la realización de un algoritmo biométrico es, sin lugar a dudas, un proceso bastante engorroso y complejo, por lo que se quiere hacer una propuesta de entorno colaborativo, donde quedarán bien definidas las tecnologías a utilizar.

1.3.4 Herramientas para el diseño de algoritmos.

Como parte del interés en el diseño de los algoritmos de forma general, se muestran algunas herramientas que sirven para este objetivo. Dichas herramientas presentan como característica general que son basadas en ambientes gráficos, donde los usuarios pueden interactuar y aprender de una forma más amena.

Herramienta PSEINT - PIPEH Pseudo Intérprete.

PSelnt está pensado para asistir a los estudiantes que se inician en la construcción de programas o algoritmos computacionales.

Características y Funcionalidades:[24]

- ✓ Presenta herramientas de edición básicas para escribir algoritmos de pseudocódigo en español.
- ✓ Permite la edición simultánea de múltiples algoritmos.
- ✓ Presenta ayudas para la escritura:
 - Autocompletado.
 - Ayudas Emergentes.
 - Plantillas de Comandos.
 - Coloreado de Sintaxis.
- ✓ Permite ejecutar el algoritmo paso a paso controlando la velocidad e inspeccionando expresiones.
- ✓ Determina y marca los errores de sintaxis y en tiempo de ejecución.
- ✓ Genera diagramas de flujo a partir del algoritmo escrito.
- ✓ Convierte el algoritmo de pseudocódigo a código C++.
- ✓ Ofrece un sistema de ayuda integrado acerca del pseudocódigo y el uso del programa (esta última, aún en construcción).
- ✓ Es multiplataforma (probado en Microsoft Windows y GNU/Linux).
- ✓ Es totalmente libre y gratuito (licencia GPL).

Herramienta Processing. [25]

Es una herramienta colaborativa que presenta un lenguaje resumido basado en Java y a su vez es un compilador de aplicaciones Java. Este programa de entorno gráfico sumamente sencillo (similar a un bloc de notas), es actualmente una de las herramientas más potentes destinadas al arte digital, completamente gratuita, y con muchas librerías disponibles. Con esta herramienta se puede generar imágenes vectoriales, imágenes de píxel, objetos bidimensionales y tridimensionales, generar interacción con estos objetos por parte del usuario con fines tan disímiles como programar un videojuego o una instalación multimedia. Además de su lenguaje básico, ofrece una extensa lista de

librerías que extienden su funcionalidad. Entre estas librerías se encuentran herramientas para trabajar con vídeo, audio, sonido MIDI, tipografía, XML, PDF, entre otros.

Herramienta JGrasp (Graphical Representations of Algorithms, Structures and Processes) [26]

Permite las representaciones gráficas de algoritmos, estructuras y procesos. Su lenguaje de desarrollo es Java y una de sus principales ventajas es la capacidad que proporciona para identificar estructuras dentro del código.

1.3.5 Herramientas de programación visual.

Herramienta Greenfoot. [27]

Es un ambiente interactivo de desarrollo Java para propósitos educacionales. Permite la creación de aplicaciones gráficas en dos dimensiones, como simulaciones y juegos. Greenfoot combina por igual un entorno de desarrollo visual con forma de rejilla bidimensional, y un editor convencional para desarrollar código al estilo tradicional, con su visor de clases, editor, compilador y ejecución.

Herramienta Alice. [28]

Su lenguaje de desarrollo es Java. Presenta un ambiente visual que permite programar el comportamiento de personajes en 3D, todo sin escribir texto sino utilizando las construcciones que ellos proporcionan. Elimina los errores de sintaxis y permite aprender sobre objetos, concurrencia, estructuras y eventos. En Alice se trabaja en un entorno en tres dimensiones en el cuál se van colocando objetos (por ejemplo un conejo o una caja). Una vez colocados los objetos, se les hace interactuar creando una especie de "corto de película". También es importante destacar que toda la programación es visual, no se escribe ni una línea de código, simplemente se arrastra con el ratón. Además está orientado a un entorno de objetos visuales en 3D.

1.3.6 Herramientas gráficas de diagramas en bloques (HGDB).

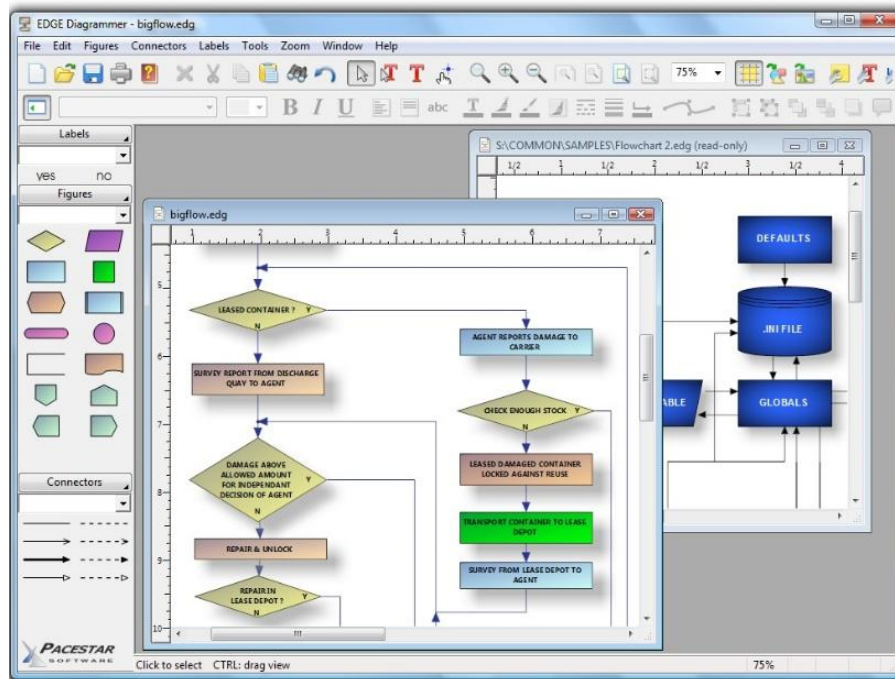


Figura 6: Ejemplo de una vista de HGDB.

Muchas herramientas de modelado gráfico han aparecido en los últimos tiempos. Dentro de esta categoría se encuentran las herramientas que utilizan los diagramas en bloques. Proporcionan muchas ventajas, ya que ahorran trabajo y tiempo a los usuarios. Una vez que estén implementados cierta cantidad de bloques, los cuales están representados de forma gráfica, es más sencillo realizar cierta actividad, ya que no es necesario tener que escribir el código. Esto también es conocido como lenguaje de programación visual. Dicho lenguaje permite a los usuarios crear programas mediante la manipulación de los elementos del programa gráfico en lugar de especificar el código. Además permite la programación de expresiones visuales, arreglos espaciales de los símbolos de texto y gráficos. [28]

Ejemplos:

Simulink: Es una herramienta gráfica de diagrama en bloques que presenta un entorno de programación visual que funciona sobre el entorno de programación Matlab. Es un entorno de más alto nivel de abstracción que el lenguaje interpretado Matlab (archivos con extensión .m). Constituye una herramienta de simulación de modelos o sistemas, con cierto grado de abstracción de los fenómenos físicos involucrados en los mismos. [29]

Harpía: Una herramienta gráfica de diagramas en bloques para la evaluación, creación y puesta en práctica de los algoritmos del tratamiento de imagen o de los sistemas de visión de computadora usando OpenCV. [30]

VisSim: Una herramienta gráfica de diagramas en bloques para la simulación de sistemas dinámicos y el diseño de modelos de base de los sistemas embebidos. Se utiliza un paradigma de flujo de datos gráficas para poner en práctica los sistemas dinámicos basados en ecuaciones diferenciales. Es ampliamente utilizado en el diseño del sistema de control y procesamiento de señal digital para la simulación multidominio y el diseño. Incluye bloques para la aritmética booleana, así como filtros digitales, funciones de transferencia y de integración numérica.[31]

1.4 Desarrollo colaborativo.

La colaboración se refiere abstractamente a todo proceso donde se involucre el trabajo de varias personas en conjunto. Trabajo donde se pone de manifiesto la cooperación y coordinación para llegar a metas comunes. La primera consiste en el trabajo en común llevado a cabo por parte de un grupo de personas o entidades mayores hacia un objetivo compartido, generalmente usando métodos también comunes, en lugar de trabajar de forma separada y en competición. La segunda es la acción de concertar medios, esfuerzos, conocimientos, entre otras cosas, para una acción común. En el caso particular de la informática se refiere a incrementar la productividad y el trabajo en equipo utilizando tecnologías de información. [32]

1.4.1 Proceso de desarrollo colaborativo.

En un proceso de desarrollo colaborativo deben participar varias personas o grupos de personas que contribuyen a lograr consensos para alcanzar un objetivo mediante acciones. Un proceso de este tipo permite combinar capacidad creativa e ideas, y elaborar contenidos que se podrían ver representados en diagramas, software, obras literarias, dibujos, pinturas o música. [33]

Los entornos colaborativos pueden proveer métodos con un proceso de desarrollo compuesto por roles, actividades y flujos de trabajo. Igualmente nos brindan herramientas de colaboración y elaboración; la colaboración se pondría de manifiesto mediante la comunicación generalmente asíncrona, dentro de los que se pueden encontrar los foros, email, correos de voz, y las aplicaciones web. Un tipo de herramientas es el software colaborativo, el cual se desglosa en herramientas de colaboración-comunicación, así como herramientas de conferencia y de gestión colaborativa que se detallan a continuación:

Las **herramientas de conferencia** se comportan de forma síncrona donde los colaboradores comparten interactivamente cierta cantidad de información. Entre los principales tipos de conferencias se encuentran: [34]

- ✓ Conferencia de Datos-PC en red, con el fin de tener en un ámbito de exposición mediante el acceso y la modificación un espacio de presentación compartido.

- ✓ Conferencia de videos, donde los colaboradores hablan y escuchan alcanzando un espacio visual-auditivo compartido.
- ✓ Conferencia inter-salas, los participantes pueden estar en una misma sala o en salas distantes, en espacios físico-visual-auditivo compartidos o reunión virtual.

Las **herramientas de gestión colaborativa** [31, 34, 35] facilitan las actividades en grupo. Entre ellas se encuentran los calendarios electrónicos, sistemas de gestión de proyectos, sistemas de control de flujos de actividades, sistemas de gestión del conocimiento y el soporte a redes sociales.

Un entorno colaborativo suele tener una plataforma legal, mediante una protección brindada por contenidos elaborados; una propiedad colectiva e intelectual, exigiendo responsabilidades con comunidades de creación, proyectos en realización y recursos administrados. Mediante las competencias y una cuantificación del esfuerzo, se logra incentivar a colaborar, logrando un ambiente de trabajo virtual colaborativo y modificable.

Cuando se habla de proceso de desarrollo colaborativo, se abarca mucho más que sólo el hecho de colaborar para alcanzar cierto objetivo común. Se abarca toda la gama de actividades, herramientas y métodos que se deben llevar a cabo para alcanzar dicho objetivo, lo que constituye el proceso como tal. Por esta razón es que en el actual epígrafe se hace un estudio de variados procesos de desarrollo colaborativo, y en diferentes ramas, para obtener una visión más amplia de sus características.

1.4.2 Entorno Colaborativo.

Los Entornos Colaborativos son lugares de trabajo donde la colaboración y el trabajo en grupos son posibles de una manera sencilla, independientemente de la localización geográfica de los desarrolladores. Así los usuarios pueden comunicarse, intercambiar información y trabajar conjuntamente. Puede haber varios a la vez que pueden expresar sus puntos de vista y reflexionar sobre las opiniones de otros. En este sentido, los entornos colaborativos extienden el lugar de trabajo más allá de los límites horarios y geográficos. [36]

Las tecnologías que sostienen el trabajo colaborativo van desde pequeñas herramientas para abordar de manera conjunta la creación de un sólo producto, como Voicethread; editores de documentos compartidos, como Adobe Buzzword, Google Docs y Etherpad, wikis y blogs, hasta entornos independientes para la colaboración como Moodle, Ning, o PageFlakes.

Entorno colaborativo virtual. [37]

Es un área virtual donde hay un conjunto de personas que desean interactuar para satisfacer sus necesidades o llevar a cabo roles específicos. Así como también comparten un propósito

determinado común. Cuenta con sistemas informáticos que median las interacciones y facilitan la cohesión entre los miembros. Para que pueda existir un entorno colaborativo virtual es necesario la existencia previa de una Comunidad Virtual, si no la comunicación e intercambio sería imposible de realizar.

Características Generales. [38, 39]

- ✓ Todos los participantes en una determinada tarea aportan sus ideas para entre todos, modelar el resultado final, de una forma consensuada, natural y lógica.
- ✓ Desarrollo descentralizado.
- ✓ Uso de diversas herramientas de comunicación asíncronas.
- ✓ Asignación de roles.
- ✓ Múltiples colaboradores con competencias diversas.
- ✓ Generalmente la colaboración es de carácter voluntario.
- ✓ El desarrollo debe realizarse a código libre.

Entorno Colaborativo para la Gestión del conocimiento. [35]

En los últimos años está cobrando relativa importancia las aplicaciones diseñadas para la colaboración entre personas o grupos de personas. Este auge se puede apreciar en diferentes ámbitos de la sociedad, como puede ser el ámbito educacional o del conocimiento. A continuación se ilustran algunos ejemplos con sus particularidades más importantes.

GECOSOFT. Plataforma para la Gestión del Conocimiento. [40]

Se compone de dos herramientas de software, una aplicación cliente denominada Macosoft y un servidor denominado Servimap. Estas herramientas automatizan los procesos de creación, almacenamiento, estructuración y socialización del conocimiento en forma de Mapas Conceptuales e incluye mecanismos para importar y exportar conocimientos desde y hacia ontologías. Con la integración de Macosoft y Servimap se logra la creación de un ambiente virtual que potencia el trabajo colaborativo en el desarrollo del conocimiento.

Herramientas utilizadas.

- ✓ Macosoft: Herramienta para la elaboración de mapas conceptuales.
- ✓ Servimap: Posee una base de conocimiento que permite crear las bases para un mayor procesamiento de este, estructurado en Mapas Conceptuales desde el punto de vista computacional.

El intercambio de información se realiza utilizando XML y a través de los Servicios Web implementados en Servimap.

Ventajas.

- ✓ Logra crear un ambiente distribuido en el que se implementan los mecanismos de: elaboración, almacenamiento, organización/clasificación.
- ✓ Búsqueda de conocimiento en forma de recursos.
- ✓ El trabajo es de forma colaborativa.
- ✓ Al apoyarse en protocolo HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

Desventajas.

- ✓ Al utilizar los servicios web puede ser que su rendimiento sea bajo. Uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.

Wikipedia. [41, 42]

La Wikipedia es una colosal enciclopedia on-line, es gratuita y permite modificar sus contenidos a los lectores. De manera más enfocada, Wikipedia es el núcleo de una comunidad de usuarios que utilizan los wikis para poner en circulación sus conocimientos, en este caso, enciclopédicos. No es la única comunidad wiki que existe en la internet, pero sí la más importante. El wiki es un software especial creado en 1995 por el informático Ward Cunningham que permite crear documentos de manera colectiva, interconectados mediante hiper-enlaces. A raíz del desarrollo de este software, empezaron a popularizarse las comunidades wiki, primero en el campo informático y científico, luego en otros muchos ámbitos.

Herramienta:

- ✓ MediaWiki: Es la herramienta sobre la cual está construida la wikipedia, es una de las plataformas más utilizadas para el desarrollo de todo tipo de wiki. Es un software para wikis libre programado en el lenguaje PHP.

Ventajas:

- ✓ La más evidente es su gratuidad.
- ✓ Hay un cúmulo de información considerable al alcance de cualquiera.

- ✓ Todo lo que allí se expone es para empleo público y filantrópico.
- ✓ Pluralidad de puntos de vista (no hay una orientación, un sesgo como en el caso de Encarta).
- ✓ La rapidez con la que se actualiza la información es impresionante.

Desventajas.

- ✓ No garantiza la objetividad de la información.
- ✓ Falta de contenido audiovisual.
- ✓ En lo estudiantil, faltan ejercicios o ejemplos para lo que son las tareas escolares.

Entorno Colaborativo para la Gestión de proyectos.

En el amplio mundo de la administración es muy utilizado también el desarrollo colaborativo.

Ejemplos:

Feng Office. [43]

Feng office es un aplicativo web, que nos permite la gestión electrónica de proyectos, tareas, documentos de negocio, gestión de calendario, proveedores, clientes, administración y gerencia. En definitiva es un entorno colaborativo que puede ayudar a optimizar la gestión del negocio y la satisfacción de los clientes.

Por sus características se puede utilizar para cualquier tipo de pequeña y mediana empresa (pyme), ya que se centra en cosas comunes: agenda, gestión de proyectos y tareas, gestión de clientes. Por otra parte puede mejorar la satisfacción de los clientes, ya que pueden acceder a una parte de usuario que les permite ver el estado de las tareas y el tiempo estimado para su finalización.

Herramientas:

La aplicación integra diversas herramientas como son el correo electrónico, calendarios, documentos de texto, planillas de cálculo, presentaciones, datos de contactos y registro de horas trabajadas.

Ventajas:

- ✓ Permite asignar diferentes roles dentro de los usuarios (para gestión administrativa, gestión de proyecto).
- ✓ Presenta un entorno gráfico, sencillo y muy cuidado, a través de pestañas.
- ✓ Es rápido y estable, apenas consume recursos del servidor donde se encuentra alojado.

Doolphy. [44] [45]

Doolphy es un gestor de proyectos online que facilita la planificación, seguimiento y análisis de todos tus proyectos y tareas. Permite crear y gestionar varios proyectos y también crear y administrar varios usuarios. Desde su panel de control se puede a simple vista, consultar las tareas pendientes y ver si existen alertas.

Herramientas:

- ✓ Panel de proyectos.
- ✓ Calendario.
- ✓ Informes.

Ventajas:

- ✓ Copias de seguridad diarias y almacenadas en múltiples ubicaciones.
- ✓ Interfaz gráfica usable e intuitiva.
- ✓ Comparte información, archivos y documentos en cada proyecto entre los trabajadores asignados al mismo.
- ✓ Sin límite de usuarios.

Desventajas:

Escasa posibilidad de personalización y una versión gratuita que sólo sirve para probar el servicio.

Redmine. [45, 46]

Es una herramienta que permite la gestión de proyectos informáticos. Está escrita usando el framework Ruby on Rails. Incluye un calendario y unos diagramas de Gantt para la representación visual de la línea del tiempo de los proyectos. Es software libre y de código abierto, disponible bajo la Licencia Pública General de GNU.

Ventajas:

- ✓ Soporta múltiples proyectos.
- ✓ Roles flexibles basados en control de acceso.
- ✓ Sistema de seguimiento de errores flexible.
- ✓ Diagramas de Gantt y calendario.
- ✓ Administración de noticias, documentos y archivos.
- ✓ Fuentes web y notificaciones por correo electrónico.

- ✓ Integración SCM (Subversion, CVS, Git, Mercurial, Bazaar y Darcs).
- ✓ Soporta diferentes bases de datos (MySQL, PostgreSQL y SQLite).

Entorno Colaborativo para el Desarrollo de Software.

Existe un variado catálogo de paquetes software cuyo principal cometido es el de integrar diferentes servicios de soporte para el desarrollo de software, de manera que sean accesibles desde una interfaz web común. Estos paquetes, denominados forjas, han permitido que los desarrolladores de software puedan concentrarse en su principal cometido, que es la atención al proceso de desarrollo de sus programas. Entre estos paquetes se encuentran: [47]

Vulcano. [48, 49]

Tiene como objetivo la promoción del Software Libre para facilitar la adopción del mismo por la industria, permitiendo su extensión desde el mundo académico y crear un efecto multiplicador, sumándose al impulso que desde gobiernos y administraciones se está ofreciendo a este proceso de desarrollo de software. Durante su primer año de vida en 2006, el proyecto se ha centrado en el desarrollo de una forja de proyectos de nueva generación. En 2007, el proyecto pretende dotar al Software Libre de:

- ✓ Metodologías Ágiles que faciliten y controlen el proceso de desarrollo.
- ✓ Mecanismos y medidas de calidad para hacerlo más confiable.
- ✓ Modelos de negocio innovadores que permitan a cada vez más empresas adoptar soluciones basadas y/o desarrolladas con SW libre.
- ✓ Creación y despliegue de un Centro de Competencia que ayude a las empresas nacionales a adoptar el modelo de SW Libre como parte de su proceso industrial.

Herramientas:

Utilización de estándares (XML, DocBook, OpenDocument, XHTML).

- ✓ CMMI: Es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. Las mejores prácticas CMMI se publican en los documentos llamados modelos. En la actualidad hay tres áreas de interés cubiertas por los modelos de CMMI: Desarrollo, Adquisición y Servicios.
- ✓ OSRMT (Open Source Requirement Management Tool): Es una herramienta de código abierto para la gestión de requisitos, está implementada sobre Java.

- ✓ REM (Requisite Management): Es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software.
- ✓ DotProject (Gestión de proyectos): Es una aplicación basada en web, multiusuario, soporta varios lenguajes y es Software libre. Está programada en PHP, y utiliza MySQL como base de datos (aunque otros motores como PostgreSQL también pueden ser utilizados). La plataforma recomendada para utilizar dotProject se denomina LAMP (Linux + Apache + MySQL + PHP). De todas formas, existen binarios para instalar dotProject bajo otros sistemas operativos tales como Microsoft Windows (NT, 2000, XP) y Mac.)
- ✓ Gantt Project (Planificación de proyectos): Es un programa gratuito para la administración de proyectos utilizando diagramas de Gantt. Entre sus características están:
 - Gráficas de Gantt.
 - Gráfica de carga de recursos.
 - Reportes en PDF y HTML.
 - Permite importar documentos de MS Project.

Ventajas:

- ✓ Se centra en el desarrollo de las infraestructuras y herramientas necesarias para que una organización o comunidad de software libre pueda definir sus formas de trabajar, diseminarlas, aplicarlas y recoger información de su utilización.
- ✓ Desarrollo distribuido en comunidades abiertas.
- ✓ Vulcano con las herramientas de soporte a los procesos de desarrollo de software ayudará a los productores de software a aumentar la calidad de sus procesos de desarrollo.
- ✓ Conversiones a formatos finales (PDF).
- ✓ Gestores de contenidos (CMS) y documentales (DMS).
- ✓ Primer proyecto incubado en la plataforma INES.

Desventajas:

- ✓ Es un desarrollo llevado a cabo por una persona individual, por lo que existe el riesgo de que no sea sostenible a lo largo del tiempo.
- ✓ No existe un soporte empresarial.

- ✓ Las nuevas versiones no están planificadas ni se anuncian claramente las mejoras que serán incorporadas.
- ✓ Es posible que las nuevas versiones no sean compatibles con las anteriores.
- ✓ No es posible generar automáticamente un documento de requisitos para entregar al cliente.
- ✓ Algunas funcionalidades no han sido desarrolladas completamente y están a medias.
- ✓ La interfaz de usuario es lenta en ocasiones.

Gforge. [50]

Se trata de una de las soluciones más avanzadas, hasta la fecha, para el desarrollo de sitios de forjas. Gforge ha sido la opción base seleccionada para su mejora de cara a la implantación del repositorio de software libre para el Open Source software Observatory Repository (OSO-R). Implementa prácticamente todas las opciones que podemos pedir a una forja hoy día, con la excepción de sistemas para el control de beneficios económicos de los proyectos hospedados, dado que dicho objetivo no parece figurar entre las prioridades de sus desarrolladores.

Herramientas:

- ✓ Subversion.
- ✓ UML.
- ✓ Microsoft Visual Studio.
- ✓ Foros.
- ✓ Herramientas de gestión y monitorización de tareas.
- ✓ Alta y gestión de errores/mejoras.
- ✓ Listas de correo.
- ✓ Herramientas de control de cambios.
- ✓ Herramientas de gestión documental.

Ventajas:

- ✓ Amplio soporte para presentar su interfaz en múltiples lenguajes.
- ✓ Sobresale en los servicios de automatización de tareas de control y administración de los proyectos.

- ✓ Existe la definición de un método de creación de plug-ins, extensiones de código que permiten añadir nuevas funcionalidades a la forja.
- ✓ Existen módulos adicionales que permiten la integración de herramientas externas como MediaWiki.

Desventajas:

- ✓ No admite la ejecución de pequeños scripts del lado del servidor que permitan la automatización de ciertas acciones, fundamentalmente en la parte de presentación de la información mediante la interfaz web.
- ✓ Ofrece un soporte limitado para la integración de autenticación mediante LDAP.
- ✓ Fuerza la utilización de un formulario web HTTPS para la subida de código a la plataforma.
- ✓ Su última versión (GForge Advanced Server) no es de código abierto.

Entorno Colaborativo para las Manifestaciones artísticas y literarias.

En ámbitos tan actuales y universales como las manifestaciones culturales, en su inmensa generalidad, se pone de manifiesto una vez más que las tecnologías deslumbran al no tener límites. Existen en el mundo diversos entornos donde las personas de diferentes latitudes se pueden unir y en conjunto, de forma colaborativa, crear obras. A continuación se ilustran algunos ejemplos y sus características principales. [51, 52]

Poliedro On line. [53, 54]

Es una propuesta de colectivos de creación musical. Por este medio se vienen conformando grupos de compositores con el objetivo de producir obras electroacústicas en forma colaborativa y remota a través de la Web. Esta propuesta pone énfasis en la web como medio de almacenamiento, difusión y distribución de música.

Herramientas:

- ✓ Browser Google Chrome.
- ✓ Creative Commons.
- ✓ Editor Audacity - MP3.
- ✓ Flash Player.
- ✓ MP3 Encoder/Converter.

- ✓ Sistema de sorteos.
- ✓ YouSendit: Es una web basada en garantizar la entrega de archivos digitales, que permite a los usuarios de forma segura enviar, recibir y seguir los archivos en la demanda. Es una alternativa al envío de grandes archivos adjuntos de correo electrónico, a través de FTP, y el envío de CD o DVD o cinta adhesiva o USB Flash Drive a través de sneakernet.
- ✓ Zshare: Con Zshare puede subir archivos, imágenes, vídeos, audio y flash gratis. También puede utilizar Zshare como almacenamiento de archivos personales: copia de seguridad de datos y protección de archivos.

Ventajas:

- ✓ Potenciar el control y la modificación dinámica de las obras en forma continua.
- ✓ Concibe y produce música, que tradicionalmente ha sido utilizada en los soportes fijos de almacenamiento (LP & CD), mediante recursos que permitan modificar las muestras de audio en tiempo real por el mismo cibernauta.
- ✓ Concede así la re-elaboración de cada trabajo comúnmente denominado como re-mixes.
- ✓ Obtiene un rendimiento multiusuario de música electrónica, para que todo el mundo pueda comenzar tocando desde el primer minuto y siendo capaz de algo más complejo.

Desventajas:

- ✓ Solo puede usarse si se dispone de una conexión a Internet más o menos rápida y fiable.

Narratopedia. [55]

Proyecto colombiano con una plataforma tecnológica donde los autores pueden poner sus obras o cuentos y recibir comentarios. Por otro lado, los lectores pueden participar de los cuentos, extendiendo la narrativa o sugiriendo ideas al escritor. También hay un espacio para producir conocimientos en torno a la cibercultura, cuyo desarrollo se puede observar en este mismo proyecto.

Herramientas:

- ✓ Edición y conversión en Photoscape.
- ✓ PhotoScape: coleccionar y visualizar imágenes.
- ✓ Gimp - Software libre para edición de imágenes digitales, tanto dibujos como fotografías.
- ✓ Menu Free Mp3 converter.
- ✓ Imagen de menú IVC.

Ventajas:

- ✓ Sencillo de usar, no es necesario tener conocimientos especiales de informática, con conocimientos básicos es suficiente.
- ✓ Posibilidades de comunicación, como por ejemplo compartir fotos, artículos, trabajos, vídeos o enlaces, mantener debates, comentar los trabajos de otros e incluso evaluarlos.
- ✓ Los sistemas de gestión de contenidos, proporcionan el soporte necesario para que cualquiera pueda publicar contenidos en la Web sin gran esfuerzo.
- ✓ La complejidad técnica de estos sistemas es transparente al usuario. Incluso separan el contenido del diseño, ofreciendo distintos estilos y plantillas, para facilitar al máximo la tarea de diseñar el sitio, y permitir a los usuarios concentrarse en los contenidos.

Desventajas:

- ✓ La comunicación que brinda no sería posible sin unas herramientas tecnológicas potentes, y sobre todo, fáciles y asequibles.

Entorno Colaborativo para el Desarrollo de algoritmos biométricos.

La biometría es una de las ciencias más novedosas de la actualidad, hay muchas cosas por hacer y descubrir en dicho campo. Las empresas que desarrollan estas tecnologías establecen su proceso y crean sus herramientas de desarrollo internas con un alto nivel de protección industrial y uso exclusivo, dado que los algoritmos biométricos pueden generar ventajas competitivas. Esta investigación no tuvo acceso a ninguno de esos entornos privados pero es de suponer que los entornos deban permitir algún grado de colaboración en su proceso. La motivación principal de este trabajo es tratar de proponer alternativas a estas herramientas en la solución propuesta.

1.5 Herramientas para la generación de Reportes.

Como parte de las herramientas que se necesitan estudiar para la elaboración del Entorno Colaborativo, se encuentran las generadoras de reportes. Estas serán imprescindibles para que pueda existir una exitosa colaboración, y por consiguiente, el diseño de algoritmos de alta calidad.

Hace años los reportes se confeccionaban en formato duro, pues no existía una herramienta capaz de realizarlos. En la actualidad, el empleo del formato digital ha venido captando el interés de todos por las facilidades que ofrece y la presencia de aplicaciones que son capaces de generarlos automáticamente. Dichos programas, denominados generadores de reportes, permiten a los usuarios obtener con facilidad datos de archivos o bases de datos. Los generadores de reportes están compuestos principalmente por dos elementos básicos, un diseñador o editor de informes y un motor de generación. El diseñador de reportes es de vital importancia ya que es una herramienta capaz de llevar a cabo la personalización del diseño, brindando a los usuarios una interfaz visual para construir sus reportes, abstrayéndolo de un conocimiento profundo para crearlo. [53] [54]

Las herramientas generadoras de reportes son utilizadas para mostrar datos de los algoritmos biométricos. Los datos pueden ser de rendimiento o precisión. Esto es de gran importancia si se quiere lograr que exista colaboración en un entorno donde el principal tema es cómo desarrollar algoritmos biométricos óptimos y precisos.

A continuación se caracterizan una serie de herramientas que sirven para diseñar y generar reportes. Este estudio tiene como objetivo que se utilice la herramienta más conveniente para el Entorno Colaborativo.

1.5.1 Report Manager.

Es una aplicación de generación de reportes (Report Manager Designer). Posee una interfaz visual con buena apariencia y permite insertar componentes tales como número de página, campo de expresión, texto y etiquetas, imágenes y gráficos de barras. Incluye una librería dinámica estándar con funciones que proporcionan una interfaz con distintos lenguajes de programación como GNU C. Este sistema es un proyecto de software libre bajo el modelo MPL (incluye permiso de uso en aplicaciones GPL), por lo que puede ser usado en distintas aplicaciones incluyendo las que son de carácter comercial, pero cualquier mejora introducida en el motor de impresión debe ser publicada bajo esta licencia. Corre tanto en GNU/Linux como en Microsoft Windows, y dispone de generación en varios formatos de salida como son PDF, HTML y XSL. [55] [56]

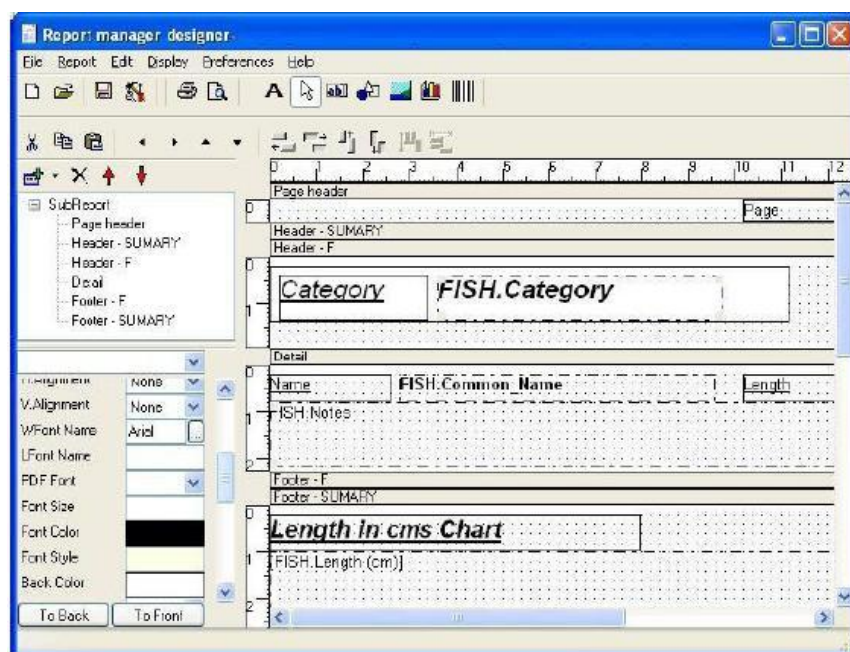


Figura 7: Pantalla de la aplicación Report Manager Designer

1.5.2 Pentaho Reporting.

Pentaho Reporting es la solución proporcionada por pentaho e integrada en su suite para el desarrollo de informes. Originariamente el proyecto se llamaba JFreeReports hasta la adquisición por parte de pentaho. Existen tres productos diferentes con enfoques distintos. Por una parte, existe un

editor basado en Eclipse con prestaciones profesionales de personalización de informes denominado Report designer destinado a desarrolladores de informes. Diseñador gráfico basado en “arrastrar y soltar” (drag & drop) que provee completo control de acceso a los datos, agrupaciones, cálculos, gráficas, formato, entre otras cosas para reportes de alta resolución. Posee opciones de salida flexibles incluyendo los populares formatos PDF, HTML, Microsoft Excel, entre otros.

Está estructurado de forma que los desarrolladores pueden acceder a sus prestaciones de forma rápida. Además, incluye un editor de consultas para facilitar la confección de los datos que serán utilizados en un informe. Forma parte de la Suite para Inteligencia de Negocios (BI) de Pentaho. [54] [57]

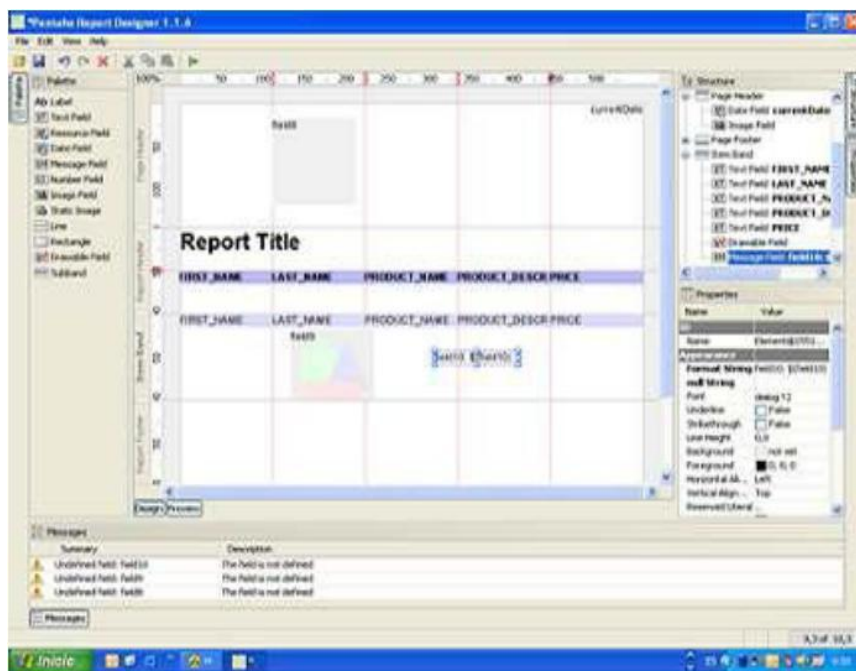


Figura 8: Vista del Report designer.

1.5.3 iReports.

La herramienta iReport es un constructor / diseñador de informes visuales muy poderoso para JasperReports escrito 100% en Java y además código abierto y gratuito. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, entre otros. Está integrado con JFreeChart, una de las bibliotecas gráficas de código abierto más difundida para Java. Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de datos, cartas, subreportes. Soporta internacionalización nativamente. Tiene asistentes para las plantillas.

iReport provee a los usuarios de JasperReports una interfaz visual para construir reportes, generar archivos “jasper” y “print” de prueba. Puede utilizarse como una herramienta de oficina para adquirir datos almacenados en una base de datos, sin pasar a través de alguna otra aplicación.

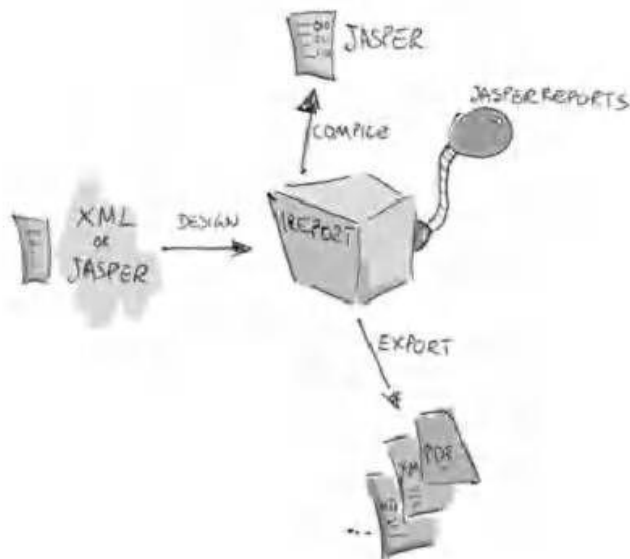


Figura 9: Funcionamiento de iReport

Puede leer y modificar ambos tipos de archivo, XML (que por convención tienen la extensión .jrxml, los cuales contienen el diseño del reporte, ver figura 11) y jasper (es el compilado de un código .jrxml, denominado .jasper). A través de JasperReports, es capaz de compilar XML a archivos jasper y “ejecutar reportes” para llenarlos usando varios tipos de fuentes de datos (JRDataSource) y exportar el resultado a PDF, HTML, XLS, CSV, entre otros. [58]

```
<!DOCTYPE jasperReport PUBLIC "-//JasperReports//DTD JasperReport//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">
<jasperReport name="report name" pageWidth="595" pageHeight="842" columnWidth="535"
leftMargin="20" rightMargin="20" topMargin="20" bottomMargin="20">
<parameter name="count" class="commonj.sdo.DataObject"/>
<background> <band/> </background>
<title> <band height="79"> ... </band> </title>
<pageHeader> <band height="35"> ... </band> </pageHeader>
<columnHeader> <band height="61"> ... </band> </columnHeader>
<detail> <band height="125"> ... </band> </detail>
<columnFooter> <band height="45"> ... </band> </columnFooter>
<pageFooter> <band height="54"> ... </band> </pageFooter>
<summary> <band height="42"> ... </band> </summary>
</jasperReport>
```

Figura 10: Muestra del código para un JRXML

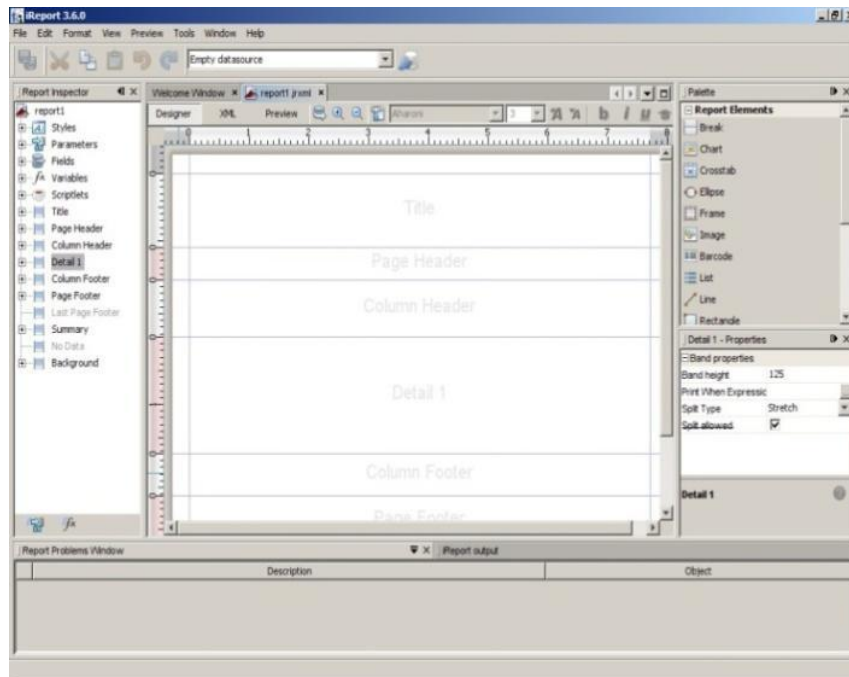


Figura 11: Vista del iReport.

1.5.4 Comparación entre los diseñadores de reportes estudiados.

Los diseñadores de reportes analizados anteriormente poseen propiedades similares a la hora de diseñar los reportes, permitiendo que se generen con un diseño personalizado y enriquecido visualmente. La mayoría de las herramientas estudiadas son de código abierto siendo esto una de las ventajas principales. También, pueden extraer datos de varias fuentes y exportar los informes en varios formatos, incluyen presentación preliminar, e incorporan varias opciones de formato.

En el caso de Report Manager, este no presenta una amplia gama de opciones en cuanto la creación de múltiples diseños. Pentaho Reporting forma parte de la Suite para Inteligencia de Negocios (BI) de Pentaho, que es orientado al desarrollador y a usuarios especializados.

Por otra parte, iReport es un diseñador visual de código libre para JasperReports escrito en Java. Es un programa que apoya a los usuarios y desarrolladores que usan la librería JasperReports para diseñar reportes visualmente, los mismos no tienen que preocuparse por conocer la sintaxis del código fuente de sus reportes para generarlos.

1.6 Herramientas para el control de versiones.

Parte de las herramientas que figuran como indispensables a la hora de pensar en el objetivo de la presente investigación, son las de control de versiones. Una herramienta de control de versiones que se encuentre adecuadamente integrada a un proceso de desarrollo colaborativo de algoritmos biométricos, es uno de los principales problemas que presenta el CISED, y una de las causas que conduce al presente trabajo.

Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas. Todos los sistemas de control de versiones se basan en disponer de un repositorio, que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados.

Clasificación de los Sistemas de Control de Versiones.

La principal clasificación que se puede establecer está basada en el almacenamiento del código: [56]

Centralizados: Existe un repositorio centralizado de todo el código, del cual es responsable un único usuario (o conjunto de ellos). Se facilitan las tareas administrativas a cambio de reducir flexibilidad, pues todas las decisiones fuertes (como crear una nueva rama) necesitan la aprobación del responsable. Algunos ejemplos son CVS y Subversion.

Distribuidos: Cada usuario tiene su propio repositorio. No es necesario tomar decisiones centralizadamente. Los distintos repositorios pueden intercambiar y mezclar revisiones entre ellos. Ejemplos: Git y Mercurial.

A continuación se describen algunas de las herramientas más utilizadas:

1.6.1 CVS (Concurrent Versions System).

Es una aplicación informática que implementa un sistema de control de versiones, mantiene el registro de todo el trabajo y los cambios en los ficheros que forman un proyecto y permite que distintos desarrolladores colaboren. CVS se ha hecho popular en el mundo del software libre. Sus desarrolladores difunden el sistema bajo la licencia GPL.

Típicamente, cliente y servidor se conectan utilizando Internet, pero con el sistema CVS el cliente y servidor pueden estar en la misma máquina. El sistema CVS tiene la tarea de mantener el registro de la historia de las versiones del programa de un proyecto solamente con desarrolladores locales. Originalmente, el servidor utilizaba un sistema operativo similar a Unix, aunque en la actualidad existen versiones de CVS en otros sistemas operativos, incluido Windows. Los clientes CVS pueden funcionar en cualquiera de los sistemas operativos más difundidos. [57]

1.6.2 Subversion.

Es un sistema de control de versiones diseñado específicamente para reemplazar a CVS. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn por ser el nombre de la herramienta utilizada en la línea de órdenes. Una característica importante de

Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente, en cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un instante determinado.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada.

Subversion presenta una serie destacable de ventajas sobre otros sistemas de control de versiones, estas son:

- ✓ Se sigue la historia de los archivos y directorios a través de copias y renombrados.
- ✓ La creación de ramas y etiquetas es una operación más eficiente. Tiene costo de complejidad constante ($O(1)$) y no lineal ($O(n)$) como en CVS.
- ✓ Se envían sólo las diferencias en ambas direcciones (en CVS siempre se envían al servidor archivos completos).
- ✓ Puede ser servido mediante Apache, sobre WebDAV/DeltaV. Esto permite que clientes WebDAV utilicen Subversion de forma transparente.
- ✓ Maneja eficientemente archivos binarios (a diferencia de CVS que los trata internamente como si fueran de texto).

Subversión también está siendo adoptado en el mundo corporativo. En un informe 2007 de Forrester Research, reconocía a Subversion como el líder destacado en la categoría de sistema de control de versiones. [58]

1.6.3 Git.

Es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

Git presenta un fuerte apoyo al desarrollo no-lineal, y también rapidez en la gestión de ramificaciones y mezclado de diferentes versiones. Incluye herramientas específicas para navegar y visualizar un historial de desarrollo no-lineal. Los almacenes de información pueden publicarse por HTTP, FTP, o mediante un protocolo nativo, que puede ser a través de una conexión TCP/IP. [59]

1.6.4 Mercurial.

Es un sistema de control de versiones multiplataforma, para desarrolladores de software. Está implementado principalmente haciendo uso del lenguaje de programación Python, pero incluye una implementación binaria de diff escrita en C. Mercurial fue escrito originalmente para funcionar sobre Linux. Ha sido adaptado para Windows, Mac OS X y la mayoría de otros sistemas tipo Unix. Mercurial es, sobre todo, un programa para la línea de comandos. Las principales metas de desarrollo de Mercurial incluyen un gran rendimiento y escalabilidad; desarrollo completamente distribuido, sin necesidad de un servidor; gestión robusta de archivos tanto de textos como binarios; y capacidades avanzadas de ramificación e integración, todo ello manteniendo sencillez conceptual. Incluye una interfaz web integrada.

Al igual que Git, Mercurial usa resúmenes para identificar revisiones. Para el acceso a repositorios mediante red, Mercurial usa un protocolo eficiente, basado en HTTP, que persigue reducir el tamaño de los datos a transferir, así como la proliferación de peticiones y conexiones nuevas. [60]

1.7 Herramientas de Gestión de Bases de Datos.

Un Sistema Gestor de Bases de Datos (SGBD) es de gran importancia para lograr los objetivos propuestos en la presente investigación. La cantidad de información que se necesita almacenar no es extensa, por lo se estudian SGBD sencillos pero buscando que sean lo más eficientes y rápidos posibles. [61]

1.7.1 PostgreSQL.

Es un sistema de gestión de base de datos relacional orientado a objetos y libre. Algunas de sus características principales es la alta concurrencia que presenta. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último que se actualizó. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos. Provee nativamente soporte para:

- ✓ Números de precisión arbitraria.
- ✓ Texto de largo ilimitado.
- ✓ Figuras geométricas (con una variedad de funciones asociadas).
- ✓ Direcciones IP (IPv4 e IPv6).
- ✓ Bloques de direcciones estilo CIDR.
- ✓ Direcciones MAC.

✓ Arreglos.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. [62]

1.7.2 SQLite.

Es un sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña biblioteca en C. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host.

Varios procesos o hilos pueden acceder a la misma base de datos sin problemas. Varios accesos de lectura pueden ser servidos en paralelo. Un acceso de escritura sólo puede ser servido si no se está sirviendo ningún otro acceso concurrentemente. En caso contrario, el acceso de escritura falla devolviendo un código de error. Esta situación de acceso concurrente podría cambiar cuando se está trabajando con tablas temporales. Este punto fue tratado en la versión 3.3.4, desarrollada el 11 de febrero de 2006. Es multiplataforma. [63]

1.7.3 MySQL.

Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante adelantar monitoreo sobre el desempeño para detectar y corregir errores tanto de SQL como de programación. Es multiplataforma. [64, 65]

1.8 Herramientas de Entorno de Desarrollo Integrado.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz

gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, entre otros. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C. [66]

A continuación se presentan algunos IDEs que pueden ser imprescindibles en la propuesta:

1.8.1 Matlab.

Es un ambiente integrado que combina computación numérica, gráficos, visualización avanzada y un lenguaje de programación de alto nivel. Posee un amplio rango de herramientas para modelar sistemas de control, análisis, simulación y procesamiento de prototipos. Presenta entre otras características:

- ✓ Cálculos intensivos desde un punto de vista numérico.
- ✓ Gráficos y visualización avanzada.
- ✓ Lenguaje de alto nivel basado en vectores, arreglos y matrices.
- ✓ Colección muy útil de funciones de aplicación.

MATLAB dispone en la actualidad de un amplio abanico de librerías de componentes de apoyo especializado, denominados Toolbox (Caja de Herramientas), que extienden significativamente el número de funciones incorporadas en el programa principal. Estos cubren en la actualidad casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el proceso de imágenes, señales, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neurales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, entre otros. Es un entorno de cálculo técnico, que se ha convertido en estándar de la industria, con capacidades no superadas en computación y visualización numérica.

Matlab cuenta con un producto llamado MATLAB Coder, el cual es muy reciente e innovador. MATLAB Coder permite a los ingenieros de diseño generar automáticamente código C y C++ legible y portable directamente desde sus algoritmos de MATLAB. Este nuevo producto elimina la necesidad de traducir manualmente código de algoritmos de MATLAB a código C y C++ para prototipado, implementación e integración de software. El resultado de esta automatización es un flujo de trabajo de desarrollo de sistemas más rápido y eficiente. MATLAB Coder admite una amplia serie de funciones del lenguaje de MATLAB para el desarrollo de algoritmos, incluidos operadores matemáticos con matrices y la gestión dinámica de matrices. Se admiten cientos de operadores, funciones y objetos de sistema, incluidos numerosos algoritmos avanzados para procesamiento digital de señal y comunicaciones. [29, 67]

1.8.2 Octave.

Octave o GNU Octave es un software para matemática diseñado por John Eaton (2007), para resolver problemas relacionados al análisis numérico. Es un software del proyecto GNU lo que implica que se encuentra dentro del software libre de licencia GPL. Es la alternativa libre a MatLab, aunque muy en desventaja debido a que no posee todas las librerías ni cajas de herramientas de este. Octave, además de ser utilizado para el cálculo numérico, se emplea en el procesamiento de señales e imágenes, estadística, geometría, redes neuronales, sistemas de control realimentados y hasta dibujo vectorial (Morales, 2004). Para graficar, Octave utiliza otro software libre llamado GNUPlot.

Octave, escrito en lenguaje C++, cuenta con una interface para el usuario del tipo línea de comando. En ella se tipean los comandos que tienen una sintaxis similar a la utilizada en MatLab. Existe una interface, más amigable con el usuario, denominada Octave Workshop, que es un ambiente integrado de desarrollo diseñado para Octave. El software GNU Octave está disponible para Linux, Windows y Mac OS. [67]

1.9 Plataforma Legal. Licencias

En todo proceso creativo, donde existen una o más personas interactuando con productos elaborados por otros individuos, es necesario establecer un mecanismo para impedir que personas mal intencionadas cometan fraude, como puede ser intentar apropiarse de la propiedad intelectual ajena. Entre estos mecanismos se encuentran las licencias.

Una licencia es un contrato entre el desarrollador de un software sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes. Es el desarrollador, o aquél a quien éste haya cedido los derechos de explotación, quien elige la licencia según la cual distribuye el software. [71]

Teniendo en cuenta que en el entorno colaborativo se van a crear software, en este caso específico algoritmos, se exponen los principales tipos de software que hay y sus características:

- ✓ Software libre [72]: Proporciona la libertad de:
 - Ejecutar el programa, para cualquier propósito;
 - Estudiar el funcionamiento del programa, y adaptarlo a sus necesidades;
 - Redistribuir copias;
 - Mejorar el programa, y poner sus mejoras a disposición del público, para beneficio de toda la comunidad.
- ✓ Software con copyleft [73]: Software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando lo redistribuyen o modifican, o sea, la versión modificada debe ser también libre.
- ✓ Software privativo: Aquél cuyo uso, redistribución o modificación están prohibidos o necesitan una autorización.

A continuación se muestran las licencias más usadas en los software no propietarios, una breve descripción y su compatibilidad con la licencia GNU (GPL):

Tabla 1: Licencias de Software Libre.

Nombre	Descripción	Compatibilidad GNU
GNU Public (GPL)	Libre, abierta, con copyleft.	Si
GNU Reducida (LGPL)	GPL sin copyleft, permite enlazar con módulos no libres.	Si
BSD Original (BSD)	Permisiva, sin copyleft, con cláusula de advertencia.	No

Otro tipo de licencia son las de software propietario o privativo. En ellas los propietarios establecen los derechos de uso, distribución, redistribución, copia, modificación, cesión y en general cualquier otra consideración que se estime necesaria. Algunos ejemplos de este tipo de licencias son las llamadas **CLUFs**: Contrato de Licencia para Usuario Final o **EULAs**: End User License Agreement, por sus siglas en Inglés. [74]

1.10 Conclusiones del capítulo

Después de todo el estudio realizado a los entornos colaborativos, los procesos de desarrollo colaborativo, los algoritmos biométricos y sus características en general, se ha llegado a la conclusión que un entorno colaborativo para algoritmos biométricos, debe ser aquel que permite integrar herramientas en una interfaz gráfica usable, que junto a procedimientos y artefactos, estén formalizados en un proceso que permita realizar de manera óptima, la colaboración de un grupo de especialistas, en el análisis de la calidad de los algoritmos, y los factores que influyen sobre la misma. Debe contar con herramientas de comunicación, donde se puedan compartir observaciones u opiniones sobre los artefactos. Para esto es importante que exista un control de versiones y una BD de prueba confiable, donde se puedan guardar dichos artefactos, así como características específicas de estos. Para obtener dichas características es necesario contar con otras herramientas que permitan medir la precisión de los algoritmos y hacer comparaciones entre ellos a través de reportes, para en caso de que sea necesario, sugerir nuevos cambios. Los algoritmos también deben ser etiquetados semánticamente con el objetivo de poder conocer dónde está lo novedoso de cada uno, y así poder hacer un análisis de estos combinados. En fin, se debe contar con un sistema que facilite el análisis de la composición de los algoritmos, para así poder detectar cuáles son las técnicas más efectivas a utilizar.

2 CAPITULO 2: PROPUESTA DE SOLUCION.

En el **Capítulo 2** se realiza la propuesta de solución, donde se define un proceso para el desarrollo de algoritmos biométricos. Se especifican varios aspectos esenciales dentro de dicho proceso, como por ejemplo las fases con las que contará y sus respectivas actividades, los roles que intervienen en cada fase o actividad, la comunicación entre los distintos roles, así como las características del ciclo de desarrollo y los artefactos generados en las distintas fases. Se propone un conjunto de herramientas las cuales serán utilizadas en las distintas fases del proceso de desarrollo colaborativo. Se definen los métodos mediante los cuales los involucrados en los distintos roles utilizarán las herramientas propuestas para ejecutar las distintas actividades del proceso propuesto.

Descripción General del Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos.

El Entorno Colaborativo está soportado por un conjunto de herramientas y un proceso de desarrollo de algoritmos biométricos, ambos orientados a lograr un desarrollo colaborativo de dichos algoritmos. Los usuarios cuentan con herramientas para diseñar algoritmos, así como para compilarlos y probarlos posteriormente. Estas herramientas están integradas con otras, como son las de almacenamiento de datos y control de versiones. Estas últimas permiten que se guarden los datos específicos de cada algoritmo así como las versiones de ellos. Además está integrada una herramienta de gestión de reportes, con la que se generan los reportes luego que los algoritmos sean probados, y los datos de sus pruebas almacenados.

El Entorno Colaborativo debe contar con un Foro, herramienta que es fundamental para la colaboración entre los distintos usuarios, debido a que proporciona la comunicación, el intercambio de ideas, de técnicas de diseño, de conocimientos, entre otras cosas.

Los usuarios tienen la opción de plasmar su derecho de autor cuando diseñen o agreguen un cambio a cierto artefacto, lo que impide el fraude intelectual. Esto es garantizado gracias a la existencia de una plataforma legal.

A continuación se presenta un diagrama donde se muestra de manera general el Entorno Colaborativo para el Desarrollo de Algoritmo Biométricos y cómo interactúan sus herramientas:

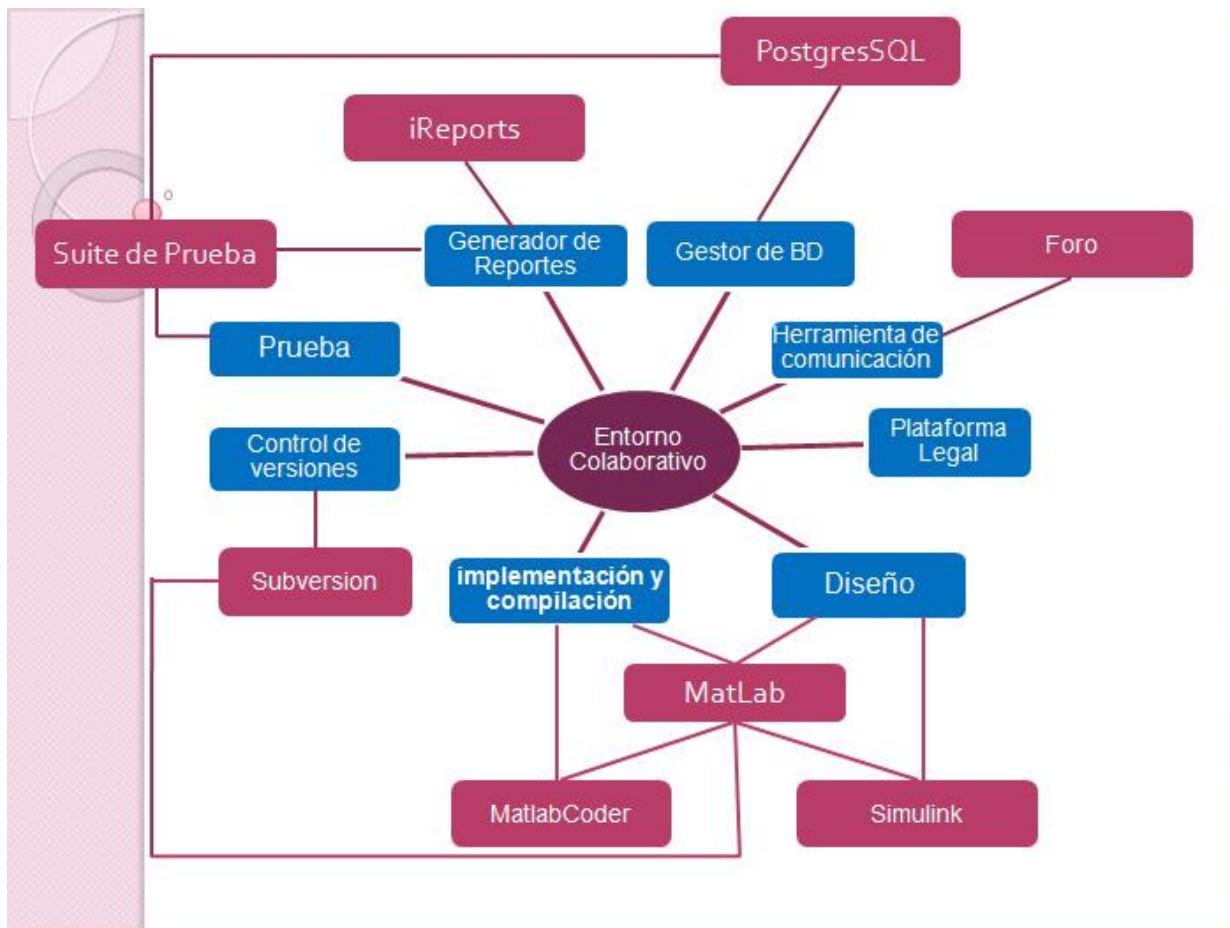


Figura 12: Descripción general del entorno.

2.1 Ciclo de Desarrollo.

Como se pudo apreciar en el Capítulo 1, la presente investigación se ha basado principalmente en los entornos colaborativos existentes en el mundo, indistintamente de la rama o el ámbito al que pertenezcan en nuestra sociedad, así como en el proceso de desarrollo de los algoritmos biométricos y las herramientas utilizadas en este. Se define que la Propuesta de Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos, va a contar con tres fases en su ciclo de desarrollo. La primera será la Fase de Diseño, la segunda la Fase de Implementación, y la tercera será la Fase de Pruebas. Dichas fases, aunque se tratarán de forma independiente, estarán estrechamente ligadas, ya que una depende de la otra.

2.1.1 Plataforma de Integración.

Basados en el análisis que se realizó sobre los Entornos Colaborativos existentes en el mundo y sus características, se propone que el proceso de integración de las herramientas se realice sobre una aplicación web. Se toma en consideración que una de las principales ambiciones es lograr la participación de una gran cantidad de usuarios en el proceso colaborativo de desarrollo de algoritmos biométricos, y las aplicaciones web presentan ventaja con respecto a esto. Otra ventaja es que estas son compatibles con cualquier plataforma, no requieren instalaciones ni configuraciones complicadas de software, las actualizaciones en todos los clientes son instantáneas, entre otras.

La aplicación web contará con una interfaz gráfica general que será la que dirija todo el proceso.

Dicha interfaz debe contar con los siguientes **Requisitos Funcionales**:

RFS1: Permitir al usuario obtener toda la información necesaria e imprescindible sobre la aplicación (Ayuda).

RFS2: Permitir al usuario diseñar un algoritmo.

RFS2.1: Permitir al usuario implementar y compilar un algoritmo diseñado.

RFS3: Permitir al usuario probar un algoritmo.

RFS4: Permitir a un usuario consultar las versiones existentes de los algoritmos biométricos.

RFS5: Permitir al usuario acceder al Foro de la comunidad.

Es importante aclarar que el **RFS5** será un requisito esencial, por lo que estará presente tanto en la interfaz principal del sistema, como en otras interfaces secundarias.

2.1.2 Fase de Diseño

En la presente fase se trata el diseño de dos tipos de algoritmos biométricos, los de extracción de características (minucias) y los de cotejo. Un buen diseño es la base de todo el proceso de desarrollo de un algoritmo. Cuando se diseña bien un algoritmo se garantiza que sea mucho más sencilla y fácil la implementación y compilación, es una de las razones por las que se propone el diseño de algoritmos biométricos utilizando tecnologías innovadoras, eficientes y que pueden facilitarle el trabajo a los implicados.

Se propone que el Entorno Colaborativo para el desarrollo de Algoritmos Biométricos cuente con una **plataforma legal**, lo que garantiza que cuando un usuario proponga un algoritmo, o modifique alguno, tenga la opción de asignarle una licencia a su producto, la que puede ser libre (GPL, LGPL, BSD) o propietaria (EULA). En caso que el usuario escoja una licencia propietaria, debe especificar cuáles son los usuarios que él quiere que tengan acceso a su algoritmo y por tanto que colaboren con él. En este caso el sistema pondrá limitaciones de acceso a los otros usuarios no especificados. Si el usuario escoge una licencia libre como es el caso de (GPL, LGPL, BSD), entonces el artefacto podrá ser accedido y modificado por todos los usuarios.

Cuando un usuario escoge la acción de diseñar un algoritmo, se le presenta una interfaz gráfica que cuenta con una serie de requisitos, y están integradas las herramientas necesarias para el diseño de los algoritmos. Se propone que dicha interfaz cuente con los siguientes requisitos funcionales:

RFD1: Escoger el tipo de algoritmo que desea diseñar (Cotejo o Minucias).

RFD2: Acceder a las versiones existentes de algoritmos.

RFD2.1: Estos estarán divididos en las siguientes categorías:

- Huellas Dactilares.
- Reconocimiento Facial.
- Firma manuscrita.
- Iris.
- Retina.
- Patrón de voz.
- Geometría de la mano.
- Termografía.
- Dinámica de Tecleo.
- Cadencia de Pasos.
- Análisis Gestual.

RFD2.2: Mostrar en una nueva ventana en forma de árbol, todas las versiones de cada uno de los algoritmos existentes en las categorías anteriormente mencionadas (según la que escoja el usuario).

RFD2.3: Mostrar en forma de *tooltip* (información adicional que se muestra cuando el usuario pasa el puntero del mouse por encima de cierto ícono) las informaciones distintivas de cada versión.

Dicho *tooltip* entre las características que debe mostrar están:

- Id del algoritmo.
- Algunas características distintivas de la versión, como puede ser algún resultado destacable en cuanto a precisión o rendimiento, alguna técnica utilizada en el diseño, secuencia de filtros aplicados, entre otras. (Características que ayuden a un mejor análisis estadístico sobre qué factores de diseño influyeron más sobre los resultados).

RFD3: Acceder a los Reportes existentes en ese momento de cada una de las versiones.

RFD4: Acceder al Foro de la Comunidad.

Actividad: Diseñar un algoritmo biométrico.

Teniendo en cuenta que se diseñarán dos tipos de algoritmos biométricos, se abordará el diseño de cada uno de forma separada, con el objetivo de propiciar una mayor comprensión.

Diseño de un algoritmo biométrico para la extracción de características (minucias).

Este tipo de algoritmo biométrico realiza en el sistema la función de procesar la imagen y extraer las características necesarias de esta. Con el estudio que se realiza a algunas herramientas gráficas de diagramas en bloques de instrucciones, se llegó a la conclusión que este tipo de tecnología es utilizado en el mundo, entre otras cosas, para el procesamiento de imágenes. Por las características de dichos algoritmos, los cuales procesan las imágenes utilizando una serie de acciones o

instrucciones que van brindando distinta información, se propone utilizar el estilo arquitectónico de Tuberías y Filtros, de forma tal que esto pueda ser aprovechado en un futuro para definir cuáles son los filtros más adecuados a utilizar en el procesamiento de cada categoría de una imagen biométrica (ya sea una imagen de huella, firma, rostro, entre otras). Pueden existir distintas frecuencias y configuraciones de filtros en dependencia del proceso a aplicar a la muestra de cada tipo de biometría. Actualmente existen categorías de las que no se tiene información en el CISED de cuáles son los filtros más apropiados.

Herramienta para diseñar un algoritmo biométrico (minucias).

Por las características de estos algoritmos, es decir, por la función que realizan y cómo la realizan, se propone utilizar una herramienta que permita graficar diagramas de instrucciones en bloques, esto es de gran ayuda visual para los diseñadores y favorece la utilización del estilo arquitectónico propuesto en el epígrafe anterior, donde cada bloque de instrucciones sería un filtro. Dicha herramienta es el Simulink, la cual está integrada al Matlab y presenta variedad de opciones necesarias para el tipo de diseño propuesto. Se propone la implementación de una serie de bloques biométricos, que tendrían como nombre "Biometrics Blockset". Es necesario que el código que tengan como fuente estos bloques sea un lenguaje con extensión .m (código Matlab). Esto quiere decir que deben ser implementados en dicho lenguaje, con el objetivo que todos los diseños de algoritmos tengan el mismo código base.

Por estudios precedentes realizados en el Departamento de Biometría del CISED, se han llegado a definir algunos de los bloques de instrucciones que van a ser necesarios implementar en la herramienta propuesta. Dichos bloques (filtros) son:

Tabla 2: Filtros propuestos

Categoría	Binari- zación	Segmen- tación	Esquele- tización	Etiquetado de Trazos	Caja de Feret	Norma- lización	Cálculo de Distancias	Filtro de Gabor	Determinar Minucias
Firma	X	X	X	X	X	X	X		
Huella			X			X		X	X

Diseño de un algoritmo biométrico de cotejo (Coincidencia).

Este es el algoritmo que, utilizando las características brindadas por el algoritmo de minucias, realiza la comparación con determinadas características que tiene guardadas en la base de datos de referencia, arrojando como resultado hasta qué punto una persona es quien dice ser. Este tipo de algoritmos se caracterizan por ser implementados basándose en operaciones matemáticas complejas, como es el uso de vectores y matrices.

Herramienta para diseñar un algoritmo biométrico de cotejo (Coincidencia).

Por las características de dichos algoritmos, se propone utilizar Matlab para el diseño de estos. Matlab cuenta con muchas tecnologías necesarias para diseñar algoritmos netamente matemáticos.

Roles que intervienen en la fase.

Basándose en lo que significa el trabajo colaborativo, se hace necesario la participación de un gran número de personas en dicha actividad. Todo usuario que tenga interés en investigar y aprender del tema tratado, puede aportar conocimientos e ideas importantes a la hora de desarrollar un algoritmo, y ese es el objetivo del presente entorno, fomentar el desarrollo colaborativo. Se propone que los usuarios que participen en esta fase tengan el rol de Diseñador.

Artefacto generado en la fase.

El artefacto generado en esta fase es el diseño de un algoritmo biométrico, ya sea de extracción de características o de cotejo. El diseño es el algoritmo expresado en el código matemático Matlab (extensión .m), antes de ser transformado hacia un lenguaje final.

2.1.3 Fase de Implementación.

La presente fase es de vital importancia, los algoritmos diseñados deben ser llevados a un lenguaje final, es decir que deben ser implementados y compilados, de forma tal que queden listos para las pruebas. En la fase de diseño se prioriza el hecho de que los algoritmos, independientemente de las técnicas usadas en cada uno de ellos, al final estén diseñados utilizando como código base el mismo lenguaje, en este caso el lenguaje matemático Matlab, por todas sus características y ventajas en cuanto operaciones matemáticas complejas se refiere. Esto hace menos traumático las transformaciones necesarias en los diseños de algoritmos biométricos.

Actividad: Implementación y compilación de un diseño de algoritmo biométrico.

En esta actividad se cogen los algoritmos biométricos previamente diseñados y se transforman hacia un código final, con el objetivo de que estén listos para ser probados y analizados por la comunidad.

Herramientas para la implementación y compilación de un algoritmo biométrico.

Una vez que ambos tipos de algoritmos biométricos estén diseñados con el mismo lenguaje base (.m), se propone como herramienta Matlab para generar el código final. Matlab cuenta con las herramientas necesarias para generar automáticamente código C++. Para aprovechar al máximo dichas posibilidades, se propone que el lenguaje final de los algoritmos sea C++. Esto sin lugar a dudas hace mucho más sencillo el trabajo de los diseñadores de algoritmos, lo que agiliza el proceso propuesto.

Roles que intervienen en la fase.

La propuesta que se hace en el presente trabajo de las herramientas a utilizar en la implementación y compilación de los algoritmos diseñados, tiene como objetivo facilitarles el trabajo a los diseñadores.

De esta forma no tendrán que pasarse, semanas o incluso meses, traduciendo lenguajes que resultan complicados por su nivel de complejidad hacia cierto lenguaje, en este caso C++. Por esa razón se han aprovechado al máximo las ventajas proporcionadas por dichas herramientas. La idea que se persigue es que cada diseñador, cuando acabe el diseño de cierto algoritmo, genere el código pertinente inmediatamente y sean guardados tanto el código (listo para pruebas), como el diseño del algoritmo. Después que el código C++ esté generado, se propone que el usuario evalúe ese código en la suite de prueba, con el objetivo de comprobar que el resultado de su modificación no llevó consigo un impacto en la precisión ni en el método matemático usado, y las operaciones de C++ actualizadas son equivalentes a las del modelo matemático. Luego que el usuario haya evaluado su versión del algoritmo, puede realizar algún cambio en el código C++, buscando la optimización del mismo si lo cree necesario. Se propone que los usuarios que participen en esta fase tengan el rol de Programador.

Artefacto generado en la fase.

Los algoritmos al ser implementados y compilados llegan expresados como código fuente (C++), y están listos para ser probados en la Suite de pruebas. El artefacto generado en esta fase es una aplicación compilada y lista para probar.

2.1.4 Fase de Pruebas.

La tercera fase del presente trabajo es la de Pruebas, las cuales van a tener un papel determinante en el presente Entorno Colaborativo; son las que brindan la información a toda la comunidad de hasta qué punto son precisos y óptimos los algoritmos diseñados por los usuarios. Es importante realizar la mayor cantidad de pruebas y que a su vez sean lo más precisas posible, con el objetivo que aumente el rango de debilidades de cada artefacto (algoritmo).

Cuando el usuario seleccione la opción Realizar Pruebas, el entorno muestra una interfaz gráfica general de pruebas que contiene los **Requisitos Funcionales** que se definen a continuación:

RFP1: Acceder a la Suite de Prueba.

RFP2: Acceder a las versiones existentes de algoritmos.

RFP3: Acceder al Foro de la Comunidad.

RFP4: Acceder a los Reportes existentes en ese momento.

Que se les permita a los usuarios realizar todas las acciones descritas en los requisitos funcionales hace más flexible y explícito la actividad de probar un algoritmo. Esto posibilita que un usuario, si desea, pruebe cualquier algoritmo que le haya llamado la atención por alguna razón en específico. Lo mismo sucede con los reportes. El usuario puede acceder a todos los reportes si es de su interés, de manera tal que pueda buscar datos, estadísticas y gráficas comparativas entre los algoritmos y cada una de sus respectivas versiones. Al igual que en las fases anteriores, en esta también se puede

acceder al Foro de la Comunidad, que permite a los usuarios expresar sus opiniones e intercambiar criterios.

La Fase de Pruebas está conformada por una serie de actividades, que son las que propician el desarrollo de las pruebas en un ambiente de colaboración. Se definen las siguientes actividades:

Actividad 1: Probar un algoritmo.

A diferencia de otros procesos de desarrollo colaborativo, específicamente para el de algoritmos biométricos se cuenta con la ventaja de poder cuantificar la calidad de los artefactos obtenidos con el uso de las métricas que cuantifican tanto la precisión como el rendimiento. A continuación se muestran algunas de estas métricas existentes.

Precisión:

- ✓ Tasa de Falsos Aceptados (FAR) (FMR).
- ✓ Tasa de falsos rechazados (FRR) (FNMR).
- ✓ Tasa de error igual (EER).
- ✓ ZeroFMR (CeroFAR).
- ✓ ZeroFNMR (CeroFRR).
- ✓ FMR100 (FAR100).
- ✓ FMR1000 (FAR1000).
- ✓ REJ_{ENROLL} .
- ✓ REJ_{MATCH} .

Rendimiento:

- ✓ (AET) (Average enrollment time).
- ✓ (AMT) (Average matches time).
- ✓ (ATS) (Average template size).
- ✓ (MTS) (Maximum template size).
- ✓ (MEM) (Maximum enrollment memory).
- ✓ (MMM) (Maximum matches memory).

Herramienta para probar un algoritmo.

SUITE DE PRUEBAS.

Para que una herramienta de pruebas cumpla con todo lo necesario para diseñar algoritmos biométricos que tengan realmente una calidad alta, debe ser capaz de:

- Brindar todos los datos anteriormente descritos sobre los algoritmos.
- Teniendo en cuenta los dos tipos de algoritmos biométricos que se estudian:
 - Los que procesan la imagen y extraen las características (minucias).
 - Los de cotejo (comparación).

La Suite de Pruebas debe permitir que un usuario pruebe ambos de forma independiente, esto hace el trabajo más sencillo a la hora de diseñar un algoritmo. Permite que el usuario se concentre (y perfeccione más) en cierto tema específico de un algoritmo.

- Permitir que el usuario pueda realizar pruebas que arrojen tanto resultados positivos como negativos. (La Suite debe contar con una gran variedad de **Bases de Datos** de referencia).
 - El sistema de pruebas de un Entorno Colaborativo para el desarrollo de algoritmos biométricos debe tener una Base de Datos que presente distintos juegos de prueba, entre esos juegos deben existir algunos que brinden resultados positivos y otros negativos, de forma tal que las pruebas sean más profundas. Dichos juegos están separados por cada categoría biométrica (huella, firma, entre otras.)
 - En esta Base de Datos se deben guardar los Reportes de los algoritmos que ya han sido probados. Para esto es necesario que las tablas estadísticas estén formadas por los identificadores (Id) de los algoritmos (estos Ids deben ser los mismo con los que están guardados los algoritmos en el sistema de control de versiones), y los resultados de las pruebas de cada uno de ellos. La forma en que se van a generar dichos resultados se explicará en epígrafes posteriores.

Para lograr esto son necesarios los siguientes **Requisitos Funcionales**:

RFP1: Crear pruebas.

- 1.1. Cargar persona.
- 1.2. Cargar base de datos de referencia.
- 1.3. Cargar estados de las pruebas.
- 1.4. Cargar la categoría (tipos de biometría: huellas, rostros iris, otras.)

RFP2: Cargar algoritmo biométrico.

- 2.1. El cliente selecciona la base de datos de referencia con la que va a probar su algoritmo biométrico.

2.2. El cliente carga la aplicación biométrica (es el fichero que contiene al algoritmo).

RF3: Guardar datos de las pruebas realizadas.

3.1. El probador después que realiza la prueba almacena los resultados en la base de datos.

3.2. Los datos pueden ser visualizados posteriormente por el Diseñador de Reportes, para que este pueda generar los reportes pertinentes.

Roles que intervienen en la actividad.

Con la presente propuesta se busca lograr mayor rapidez y calidad en todos los procesos. Cada actividad debe realizarse de forma dinámica para poder lograr dichos objetivos. Por lo tanto se propone que cada usuario pueda probar un algoritmo, que no tiene que ser necesariamente uno diseñado por él. Se propone que los usuarios tengan los roles de Probador.

Actividad 2: Generación de reportes de resultados.

Luego que se haya realizado la actividad de probar uno o muchos algoritmos, entonces se realiza el diseño y generación de reportes, para luego publicar los datos de cada algoritmo en los formatos definidos.

Los reportes siempre son generados de forma tal que sugieren comparaciones entre los algoritmos. El Diseñador de Reportes debe recopilar todos los datos brindados por la Suite de Pruebas después de hacerle la prueba a cierto algoritmo, y generar el reporte, tanto en forma de tablas estadísticas como en forma de gráficas, con el objetivo de hacer comparaciones estadísticas entre todas las versiones de los algoritmos existentes. Esto incrementa en un gran por ciento el nivel de entendimiento de todos los que se encuentran involucrados en el proceso de colaboración.

- Las tablas estadísticas están formadas por los algoritmos y todos los parámetros mencionados anteriormente. Un ejemplo de cómo son los reportes es la siguiente tabla generada por Fingerprint Verification Competition (FVC 2006):

Tabla 3: Tabla estadística comparativa donde P017,....., P072, son algoritmos. (Según el FVC 2006).

Alg	EER	FMR 100	FMR 1000	Zero FMR	REJENROLL	REJMATCH	Avg Enroll Time	Avg Match Time	Avg Model Size	Max Model Size	Max Enroll Mem	Max Match Mem
P017	5.564%	9.708%	15.335%	22.922%	0.00%	0.00%	0.038s	0.039s	1.22Kb	1.66Kb	1472Kb	2172Kb
P066	5.978%	9.556%	14.167%	19.405%	0.00%	0.00%	0.430s	0.506s	5.63Kb	10.84Kb	8080Kb	8520Kb
P045	6.122%	10.498%	22.348%	41.494%	0.00%	0.00%	0.074s	0.311s	3.88Kb	5.23Kb	1476Kb	1916Kb
P131	6.922%	8.712%	21.299%	29.675%	0.00%	0.00%	0.068s	0.205s	8.02Kb	11.91Kb	5772Kb	6724Kb
P067	7.044%	12.435%	15.325%	24.177%	0.00%	0.00%	0.108s	0.120s	24.01Kb	25.37Kb	3404Kb	3600Kb
P009	7.370%	10.584%	13.344%	20.303%	0.00%	0.00%	0.265s	0.304s	1.43Kb	2.10Kb	4028Kb	4208Kb
P058	7.496%	10.779%	13.041%	15.671%	0.00%	0.00%	0.103s	0.103s	2.07Kb	4.22Kb	1636Kb	1660Kb
P074	7.733%	12.619%	17.576%	28.701%	0.00%	0.00%	0.092s	0.094s	2.04Kb	3.52Kb	1468Kb	1480Kb
P015	7.823%	11.201%	14.156%	17.814%	0.00%	0.00%	0.572s	0.597s	1.31Kb	3.26Kb	14264Kb	20324Kb
P101	7.928%	12.424%	57.413%	57.413%	0.00%	0.00%	0.176s	0.283s	4.16Kb	4.84Kb	1856Kb	7496Kb
P024	8.255%	10.898%	13.669%	16.926%	0.00%	0.00%	0.065s	0.067s	1.01Kb	1.37Kb	2100Kb	2360Kb
P088	8.794%	13.864%	18.431%	26.840%	0.00%	0.00%	0.503s	0.515s	0.89Kb	3.24Kb	1832Kb	4288Kb
P072	8.887%	13.247%	17.792%	23.853%	0.00%	0.00%	0.024s	0.035s	1.38Kb	1.38Kb	724Kb	760Kb
P060	9.124%	21.190%	51.115%	79.513%	0.00%	0.00%	0.134s	0.136s	0.34Kb	0.73Kb	1176Kb	1184Kb
P041	9.468%	12.641%	16.061%	20.444%	0.00%	0.00%	0.033s	0.049s	0.50Kb	0.68Kb	1808Kb	2308Kb

➤ Después de un estudio realizado, se concluye que lo más óptimo es generar las gráficas divididas en dos categorías, unas en cuanto a Precisión, y otras en cuanto a Rendimiento. Esto se hace necesario para que no se mezclen ambos tipos de métricas en el momento de realizar un análisis comparativo; además, de esta forma las gráficas no parecen saturadas de información, lo que a la larga mejora la comprensión, que es el objetivo de estas.

Las gráficas de Precisión (ver ejemplo Figura 13) muestran el comportamiento de los algoritmos en cuanto a:

- ✓ FAR.
- ✓ FRR.
- ✓ ERR.
- ✓ FMR100.
- ✓ FMR1000.
- ✓ ZeroFMR (CeroFAR).
- ✓ ZeroFNMR (CeroFRR).

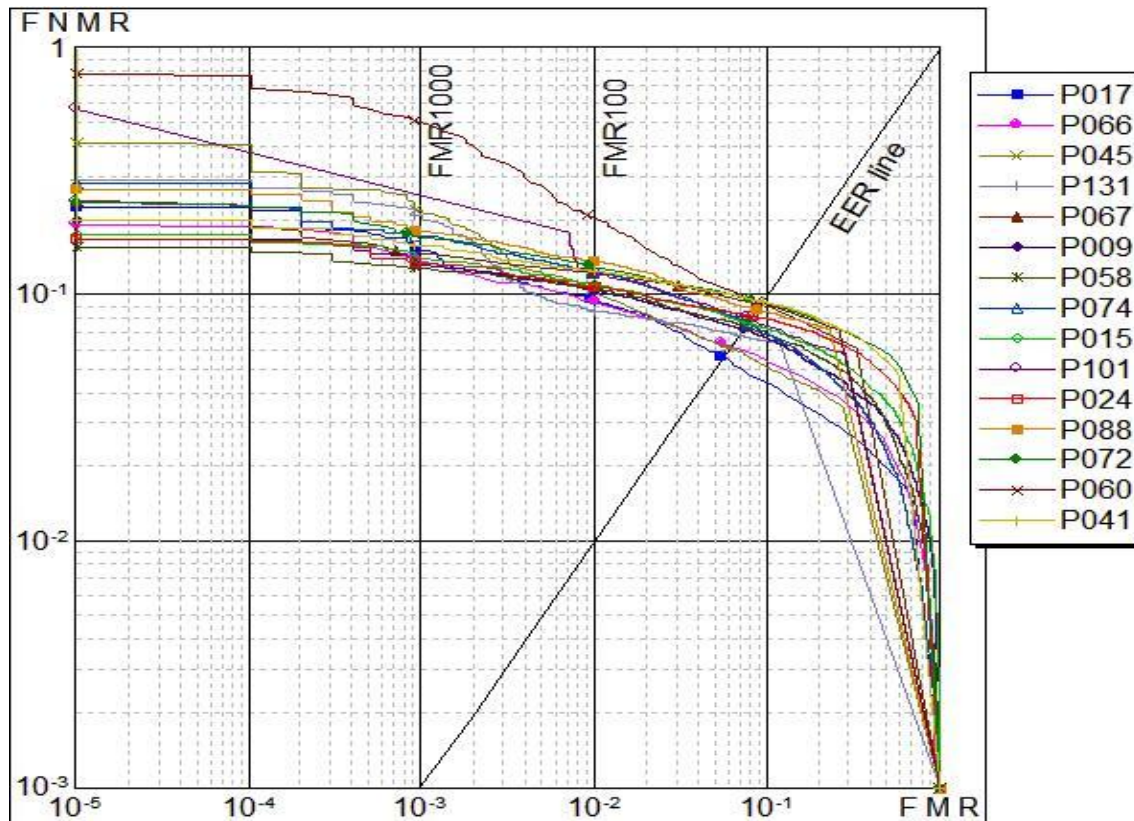


Figura 13: Se ve el comportamiento de los algoritmos: ERR,..., FRR. (Según el FVC 2006).

Las gráficas de **Rendimiento** muestran el comportamiento de los algoritmos en cuanto a:

- ✓ (AET) (Average enrollment time).
- ✓ (AMT) (Average matches time).
- ✓ (ATS) (Average template size).
- ✓ (MTS) (Maximum template size).
- ✓ (MEM) (Maximum enrollment memory).
- ✓ (MMM) (Maximum matches memory).

Herramienta para diseñar y generar Reportes de Resultados.

Se propone como herramienta para el diseño y generación de reportes JasperReports. El diseñador gráfico de esta herramienta es iReports. Dicha herramienta cuenta con todos los requisitos que se exigen para diseñar los reportes especificados, además presenta la gran ventaja de ser una herramienta libre.

Roles que intervienen.

En el caso de los reportes es necesario que exista uno o más usuarios especializados en la generación de estos. En este caso no se puede permitir que cualquier persona diseñe un reporte, ya que existen ciertos requisitos y plantillas para diseñar estos, por lo tanto debe haber personas especializadas en este tema. Se propone que dichos usuarios tengan el rol de Diseñador de Reportes.

Artefacto generado en la Fase.

Al finalizar la fase de Pruebas, se obtiene como artefacto un Reporte que está conformado por tablas y gráficas que facilitan la comparación y el análisis de los algoritmos biométricos.

2.1.5 Herramientas colaborativas.

En los epígrafes anteriores del presente capítulo se estudian las tres fases con las que cuenta el ciclo de desarrollo de un algoritmo biométrico, así como las herramientas que son necesarias de forma particular en cada una de ellas. Ahora se proponen otras herramientas que son necesarias en todo momento del ciclo de desarrollo, e imprescindibles para lograr un marco de trabajo colaborativo.

En el Capítulo 1 se estudian Entornos Colaborativos de diferentes ramas de la sociedad, se puede apreciar como la gran mayoría de ellos tienen integradas herramientas de comunicación (Foro, Chat, Correo electrónico, entre otras.). Contando con las características de nuestra universidad, en la que ya los usuarios cuentan con correo electrónico y mensajería instantánea (chat), se propone incluir en el proceso de desarrollo un Foro que está dedicado a discutir todos los temas referidos a los algoritmos biométricos y su desarrollo. De esta forma se garantiza parte de la colaboración entre los usuarios. Los usuarios tendrán la posibilidad de confrontar ideas, dudas, conocimientos, entre otras cosas. Pero esto no será posible sin la existencia de otras herramientas que son las que propician los datos y temas de discusión. Aquí es donde entra a ocupar un papel determinante una herramienta que permita guardar cada versión de los algoritmos junto a sus respectivas características.

Se propone utilizar como herramienta de control de versiones Subversion, por ser libre y presentar muchas opciones que fomentan la colaboración, una de ellas es la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones. Se destaca como dato importante que cada algoritmo y sus respectivas versiones se guardan en el sistema de control versiones, y es necesario que ellos tengan un identificador (Id) por el que puedan ser diferenciados sus datos en otros momentos del proceso de desarrollo. Los algoritmos guardados en el control de versiones deben tener tags semánticos, con las características distintivas de cada uno de ellos (estos tags son los que se muestran en forma de tooltip).

Aparte de los algoritmos y sus versiones, existen otros artefactos e información de mucha valía para el proceso de colaboración que necesitan tener constancia en el sistema. Para esto se necesita un gestor de base de datos, en la que son guardados los reportes de las pruebas que se le realizan a los algoritmos y las bases de datos de referencias que se toman como medio para probar estos. Es preciso aclarar que todo el proceso gira en torno a una versión de un algoritmo (artefacto). Toda la información que brinda la base de datos está disponible desde cualquier lugar del sistema, es decir, que el usuario esté en la fase que esté, tendrá la oportunidad de consultar, por ejemplo, los reportes de cierto algoritmo. En la base de datos se identifican los datos de los algoritmos mediante el ID de estos, que son el mismo ID que tienen en las versiones. Es conveniente que dicha BD esté integrada a la Suite de pruebas, lo que significa que la base de datos que va a utilizar dicha Suite, es la misma

que se utiliza de forma general en todo el sistema. Se propone utilizar como Gestor de Base de Datos PostgreSQL, ya que es un software libre, presenta las opciones para satisfacer los requerimientos de la propuesta y además es uno de los más utilizados en nuestra Universidad.

2.2 Conclusiones del Capítulo

En este capítulo se realiza la propuesta de solución al problema planteado. Para esto se definen las fases de desarrollo de un algoritmo biométrico, lo que sienta las bases para la creación de un proceso basado en la colaboración, de forma tal que los usuarios sean los principales protagonistas. Además se definen las tecnologías que forman parte de dicho proceso, el cual constituye el núcleo del Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos. El Entorno Colaborativo permite a los usuarios consultar toda la información necesaria para que pueda existir una colaboración y una posterior interacción entre ellos, logrando con esto alcanzar un alto nivel de conocimientos que son de beneficiosos para la producción de algoritmos biométricos en el CISED.

3 Capítulo 3: Validación de la Propuesta por Valoración de Expertos

En el capítulo anterior se realiza la propuesta de un Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos. En este capítulo se hace la validación de la propuesta por un comité de expertos debido a que la estrategia no se ha implementado en el CISED y por ende no se pueden mostrar resultados tangibles del mismo. Para la evaluación técnica de la propuesta se empleará un método cuantitativo que tiene como fundamento la evaluación por parte de expertos en el tema de criterios definidos. A lo largo de este capítulo se describe el método a emplear y se exponen los resultados derivados de la evaluación.

3.1 Validación de la propuesta.

Para la validación y aceptación de la propuesta del Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos se emplea como herramienta el panel de expertos. Este panel se conforma por especialistas del Centro de Biometría del CISED, que dominan los principales aspectos de los temas que se tratan en la investigación. A continuación se realiza una descripción del proceso de selección del panel de expertos, la elaboración de la encuesta y los resultados arrojados.

3.1.1 Panel de Expertos.

Un panel de expertos está integrado por empleados de alto nivel y especialistas en diversos puestos de trabajo con una clara visión de futuro. El objetivo de este panel es la evaluación y aprobación de la propuesta de solución que se expone en la investigación. Además, debe determinar si la propuesta puede ser implementada en el Departamento de Biometría del CISED y ser vinculada a los proyectos que se desarrollan en esta entidad, implicando esto la obtención de mejores resultados.

3.1.2 Selección del panel de expertos.

La selección del panel de expertos se realiza de forma rápida, donde se cuenta con una lista de expertos reconocidos en el campo de la Biometría y se asegura que cada uno goce de independencia para evaluar la propuesta. Normalmente, en la selección del panel de expertos se deben tener en cuenta las particularidades de cada uno, pues estas influyen decisivamente en la confianza de los resultados que se logran.

Estas características son: calificación técnica, capacidad de emitir una decisión al respecto, conocimientos específicos sobre el tema a evaluar, disposición a participar.

Seleccionar los expertos, atendiendo a las características mencionadas, ayuda a alcanzar resultados con calidad. Otros valores a tener en cuenta para la elección podrían ser: la seriedad, la honestidad, la sinceridad, la responsabilidad y otros en este sentido, que hacen que las opiniones ofrecidas sean confiables y efectivas para el objetivo planteado. En el desarrollo de este proceso se tienen en cuenta las siguientes etapas:

Determinar la cantidad de expertos: se dispuso contar con un número de 7 expertos, atendiendo al nivel de complejidad y profundidad del contenido.

Conformar el listado de los expertos: La elaboración del listado de expertos se efectuó de acuerdo a la posibilidad real de participación de los elegidos. Se estableció que los expertos a consultar deben ser profesionales, con conocimientos y experiencia en temas relacionados con algoritmos biométricos y el proceso de desarrollo de estos, para que puedan expresar un criterio efectivo sobre la propuesta.

Confirmar la participación de los expertos: Una vez elaborado el listado, se le explicó a cada uno en qué consistía el trabajo en general, la propuesta a evaluar, el objetivo de la realización de la encuesta y el plazo de entrega. Una vez recibida la respuesta positiva, se estableció el listado final de los expertos, informando a cada especialista su inclusión en el proceso a evaluar y las instrucciones necesarias para contestar las preguntas. De esta forma culmina el proceso de selección, consiguiendo la participación de los expertos elegidos.

3.1.3 Elaboración de la encuesta.

En la encuesta o guía para la evaluación de la propuesta (Anexo 1) se incluyeron varias preguntas divididas en cinco grupos de criterios, se pidió una evaluación del procedimiento en base a los mismos que se catalogan en: criterios de impacto, de implantación, de méritos científicos, de generalización y de flexibilidad. Al final del documento se da la opción a los expertos de emitir sus sugerencias o una opinión general de la solución presentada, ofreciéndoles la posibilidad de expresarse con libertad, a favor o en contra del procedimiento propuesto.

La encuesta se llevó a cabo de forma anónima. Se aplicó a cada experto personalmente, explicando detalladamente a cada uno los objetivos y resultados de la propuesta y se les dio un plazo de tiempo para entregarla.

3.1.4 Guía para la evaluación de la propuesta.

Se realiza la descripción de cada uno de los pasos que se llevaron a cabo para la validación de la propuesta de un Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos, a partir de la selección de un grupo de expertos que serán los responsables de la evaluación de la investigación y de los instrumentos que se proponen para llevar a cabo la estrategia. Cada uno de estos pasos se describe a continuación.

1. Se elabora un grupo de criterios de evaluación de acuerdo a las características de la propuesta que se realizó en este trabajo, para que los expertos emitan su opinión acerca de la misma. [Anexo 1]
2. Se organiza un comité de expertos con una cantidad mínima de 7 teniendo en cuenta su especialidad (Biometría), grado científico (Ingeniero, Máster o Doctor) y currículum (Al menos 1 año y medio de experiencia). Los resultados de las encuestas permiten realizar una evaluación cualitativa de la propuesta en excelente, buena, aceptable, cuestionable y mala.

También se da la posibilidad de emitir su opinión haciendo una valoración final de la propuesta, proponiendo todas aquellas consideraciones que estimaron convenientes. [Anexo 1]

3. Después de recibir los valores del peso relativo de cada criterio, se construye la tabla de los pesos otorgados. A continuación se muestran los pesos otorgados por los expertos en la Tabla 1 donde:

E: Es el número de expertos que realizan la evaluación

C: Es el número de criterios que son evaluados.

G: Es el número del grupo al que pertenecen los criterios.

Tabla 4: Resumen de la evaluación emitida por los expertos.

G	C/E	E1	E2	E3	E4	E5	E6	E7
G1	C1	10	10	9	9	7	9	10
	C2	10	8	9	8	9	10	9
G2	C3	10	10	9	10	10	10	9
	C4	9	10	9	10	10	9	10
	C5	8	10	10	8	10	9	8
G3	C6	8	10	9	8	9	10	9
	C7	10	10	10	9	7	8	9
G4	C8	10	10	9	9	10	10	9
	C9	10	10	10	10	8	9	10
	C10	10	10	10	10	8	10	10
G5	C11	8	9	8	7	9	8	9
	C12	8	6	6	5	7	6	5

4. Se definen categorías para los pesos otorgados por los expertos.

Baja: 6 o menos

Media: 7 u 8

Alta: 9 o 10

5. Se interpretan los pesos otorgados por los expertos para cada uno de los criterios.

En el primer grupo, correspondiente a los criterios de mérito científico, se obtuvieron como resultados que el 85,7% de los expertos considera que la propuesta presenta un valor científico alto, mientras el resto considera que es medio. Con respecto a la calidad de la investigación, el 71,4% piensa que la calidad es alta, y el resto que es media.

En el grupo de criterios de implementación de arrojaron como resultados que el 100% de los expertos considera como alta, la necesidad de implementar la propuesta así como sus posibilidades de aplicación. El 57,1% considera que la propuesta tiene altas posibilidades de satisfacer las necesidades de la producción, mientras que el resto considera que las posibilidades son medias.

En el grupo de criterios de flexibilidad, el 71,4% de los encuestados considera que es alta la adaptabilidad al Entorno Colaborativo de otros proyectos productivos del CISED. Este mismo por ciento considera que la propuesta tiene un valor alto en cuanto a eficiencia y calidad. El 28,6% restante considera que la propuesta tiene un valor medio en los dos aspectos antes mencionados.

En el grupo de criterios impacto, el 100% de los expertos considera que la propuesta tiene una alta repercusión. El 85,7% expresó que la propuesta debe tener una alta aceptabilidad en los directivos del Departamento de Biometría del CISED, mientras que el 14,3% piensa que la aceptabilidad debe ser media. Los mismo por cientos se obtuvieron en cuanto al impacto de la propuesta para el área a la que está destinada.

En el grupo de criterios de generalización se arrojaron como resultados que el 42,9% de los expertos opina que la propuesta es de fácil (Alto) entendimiento, mientras que el 57,1% cree que es un poco (Medio) difícil de entender. El 28,6% considera que se necesitan ciertos (Medio) conocimientos para poner en práctica la propuesta, mientras que el 71,4% opina que es necesario ser un especialista (Baja) en la materia para poder ponerla en práctica.

Resumiendo todos los criterios, el 72,6% de los expertos emitieron pesos de categoría Alta mientras que el 21,44% ofreció pesos de categoría Media y solo el 5,96% emitieron pesos de categoría Baja. Como parte de la evaluación cualitativa, el 85,7% considera que la propuesta tiene una categoría de Excelente, mientras que el resto opina que la categoría es Buena.

3.2 Conclusiones del Capítulo

En este capítulo se realizó la validación de la propuesta. Los resultados del panel de expertos exponen una probabilidad de éxito alta, lo que valida que la propuesta se adecúa apropiadamente a las condiciones existentes en el CISED. Cuando sea aplicada puede contribuir a la solución de las dificultades existentes en el desarrollo de algoritmos biométricos en el Departamento de Biometría de dicho centro. La Propuesta de Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos fue evaluada de excelente, cumpliendo satisfactoriamente las perspectivas de la investigación.

CONCLUSIONES.

En el presente trabajo se hizo un estudio de los entornos colaborativos existentes en el mundo en diversas ramas de la sociedad, para obtener características generales de estos. Se estudió los algoritmos biométricos y sus características, así como herramientas que pueden servir para el diseño de ellos y de otros algoritmos de forma general. Además se estudiaron otras herramientas necesarias en un proceso de desarrollo colaborativo.

Se hace una propuesta de un Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos, en la que se definen muchas tecnologías necesarias para su implementación así como un proceso de desarrollo de algoritmos biométricos, ambas cosas orientadas a lograr una producción colaborativa.

Se realizó la validación de la propuesta por criterios de expertos, obteniéndose excelentes resultados, por lo tanto se cumplió con el objetivo propuesto en el trabajo.

RECOMENDACIONES.

- ✓ Se recomienda que la presente propuesta de Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos sea implementada en el Departamento de Biometría del CISED, con el fin de lograr una producción de algoritmos biométricos de alta calidad.
- ✓ Que la propuesta se expanda hacia otros departamentos del CISED, no solo el de Biometría, realizando las investigaciones necesarias para lograrlo.
- ✓ Realizar el análisis y diseño de la propuesta de solución.

REFERENCIA BIBLIOGRÁFICA

1. Jain, A.K., P.J. Flynn, and A.A. Ross, *Handbook of biometrics*. 2008: Springer-Verlag New York Inc.
2. BRENNAN, R.G. (2006) *Seguridad vs. Privacidad*.
3. Sartori, R.A.S., *Sistemas de Identificación Biométricas Modernas y Biodinámicas*. 2007.
4. FUENMAYOR, I.G., *AVANCES EN TÉCNICAS BIOMÉTRICAS Y SUS APLICACIONES EN SEGURIDAD*. 2005.
5. Maltoni, D., *Handbook of fingerprint recognition*. 2003: Springer-Verlag New York Inc.
6. Cerón, R.A., *Arquitecturas del Software*.
7. Pressman, R.S., *Ingeniería de software*. 2000.
8. Bastarrica, M.C., *Atributos de Calidad y Arquitectura del Software*. 2005, Volumen.
9. España, S., et al. *Evaluación de la Usabilidad en un Entorno de Arquitecturas Orientadas a Modelos*. 2006.
10. Mezo, B.M., T.S. Chaparro, and A.D. Heras, *Características de las empresas que utilizan Arquitectura Orientada a Servicios y de su contexto de operación*. *Journal of Information Systems and Technology Management*, 2008. **5**(2): p. 269-304.
11. Abascal, J. and R. Moriyón, *Tendencias en Interacción Persona Computador*. Inteligencia artificial. *Revista iberoamericana de inteligencia artificial*. **2002**(16): p. 9–24.
12. Cormen, T.H., *Introduction to algorithms*. 2001: The MIT press.
13. Wu, C., *Advanced feature extraction algorithms for automatic fingerprint recognition systems*. . 2007.
14. Fons, M., et al., *Codiseño Hardware-software de un Algoritmo de Matching Biométrico*. 2003, JCRA.
15. Jiménez, Á. and M. Antonia, *Evaluación biométrica de algoritmos de autenticación de firma manuscrita*. 2010.
16. Areitio, J. and T. Areitio, *Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación*.
17. Hugo, G.O.V., *SISTEMA DE RECONOCIMIENTO DE HUELLAS DACTILARES PARA EL CONTROL DE ACCESO A RECINTOS*.
18. Heidy Alina Nuevo Leon, A.M.R.C., Enrique Reyes Bermudez, Claudia Maria Lora Pomar, Yisel Delgado Mesa (2010) *ALGORITMO DE DETECCIÓN FACIAL PARA SISTEMAS DE AUTENTICACIÓN BIOMÉTRICA*. .
19. NeuroTechnology (2009) *Iris identification for PC and Web solutions, VeriEye SDK*.
20. Novara, P., *Manual Pselnt*. 2006.
21. Ben Fry, C.R., *INTRODUCCIÓN A PROCESSING*. 2001.
22. Kölling, M., *Introducción a la programación con Greenfoot* 2001.
23. Tecnológico, O., *JAVA - Entorno de desarrollo jGrasp*. 2009.
24. Tucker, M., *Jive Software*. 2008.
25. Villanueva., C., *Alice: Aprende a Programar con objetos*. 2008.
26. Rodríguez, M.G., *Introducción rápida a Matlab y Simulink para ciencia e ingeniería*. 2003: Ediciones Diaz De Santos SA.
27. Probst, G., D. de Melo, and M. Roloff. *OpenVisionBot—Pesquisa e desenvolvimento de urna plataforma robótica móvel aberta*: IEEE.
28. Alfaro, V.M., *La simulación digital como herramienta en la docencia y la investigación*. *Ingeniería*, 2000. **10**(1): p. 2.
29. Booch, G. and A.W. Brown, *Collaborative development environments*. *Advances in Computers*, 2003. **59**: p. 1-27.
30. Rice, E.H., *The Collaboration Process in Professional Development Schools: Results of a Meta-Ethnography, 1990-1998*. *Journal of Teacher Education*, 2002. **53**(1): p. 55-67.
31. Lebrún, C.A.V., *INTEGRACIÓN DE HERRAMIENTAS DE TECNOLOGÍAS DE INFORMACIÓN "PORTALES COLABORATIVOS DE TRABAJO" COMO SOPORTE EN LA ADMINISTRACIÓN DEL CONOCIMIENTO*. 2007.
32. Grau, A., *Herramientas de gestión del conocimiento*. Disponible en: www.gestiondelconocimiento.com/documentos2/america/herramientas.htm [Consultado: 10 de noviembre del 2004].

33. Gaudioso Vázquez, E., et al., *Cómo gestionar la colaboración en el Marco Lógico Colaborativo en un entorno de aprendizaje adaptativo basado en web*. Inteligencia artificial, 2004. **8**(24): p. 121-129.
34. Churchill, E.F., D.N. Snowdon, and A.J. Munro, *Collaborative virtual environments: digital places and spaces for interaction*. Educational Technology & Society, 2002. **5**(4).
35. Accino, J.A., *ÁGORA VIRTUAL: una propuesta de entorno colaborativo y de enseñanza sobre interfaces OSID*. Boletín RedIRIS, 2006.
36. Maybury, M., *Collaborative virtual environments for analysis and decision support*. Communications of the ACM, 2001. **44**(12): p. 51-54.
37. Simón, A., et al. *GECOSOFT: Un Entorno Colaborativo para la Gestión del Conocimiento con Mapas Conceptuales*. 2006.
38. GRZEGA, J., *la Wikipedia*. Onomasiology Online, 2006. **7**: p. 1-15.
39. Gomez, M.V., *Wikipedia: la enciclopedia libre*.
40. Broncano, R. (2009) *Analizamos Feng Office, ¿la oficina del futuro?*
41. Leo, J. (2010) *Doolphy, otra plataforma para gestionar proyectos y tareas en grupo*.
42. Fernández, A., *Redmine, gestión de proyectos web*. Todo linux, 2010(120): p. 32-35.
43. López, J., *Redmine, gestión de proyectos Ruby on Rails*. Todo linux, 2009(104): p. 21.
44. Tramullas, J., *Herramientas de software libre para la gestión de contenidos*. Hipertext. net, 2005. **3**.
45. Unblogenred.es, *El Proyecto Vulcano potencia el software libre de calidad en Europa*. 2010.
46. vulcano (2007) *Vulcano*.
47. Copeland, T., et al., *GForge Manual, 2006*. URL http://gforge.org/docman/view.php/1/34/gforge_manual.pdf.
48. Red, U., *Media Art Wiki. Uso de Wikis para la enseñanza interdisciplinar y multimedia del arte de los nuevos medios de comunicación en entornos virtuales de aprendizaje*. Revista de Docencia Universitaria, 2010(0).
49. González, M. (2010) *La mediación artística: un modelo de educación artística para la intervención social a través del arte*.
<http://issuu.com/> (2009) *Poliedro On Line*.
50. Luna, F.E. (2009) *Poliedro [On line]*
51. <http://narratopedia.net/>. *Narratopedia*. 2009; Available from: <http://narratopedia.net/>.
52. Zavala, M.R. (2010) *Generación de reportes*.
53. (2010) *Manual Generador de Reportes de Pentaho(Pentaho Report Designer)*.
54. MSDN (2008) *Report Manager*.
55. reportman.sourceforge.net, *Report Manager*. 2010.
56. Gravatar (2011) *El Proyecto Pentaho BI*.
57. Herrera, C. (2005) *Introducción a iReport*.
58. Calderón, J.D. and A. de Servidores, *CAPACITACION DE SISTEMA DE CONTROL DE VERSIONES BAZAAR*.
59. PROUDHON, P.J., *Concurrent Versions System*. Software Studies: p. 64.
60. Collins-Sussman, B., B.W. Fitzpatrick, and C.M. Pilato, *Control de versiones con Subversion*. 2002.
61. García Domínguez, A., *Introducción al Sistema de Control de Versiones Distribuido Git*. 2011.
62. López, J., *Mercurial, el sistema de control de versiones distribuido*. Todo Linux: la revista mensual para entusiastas de GNU/Linux, 2009. **9**(104): p. 46-49.
63. Connolly, T.M.a.C.E.B., *Sistemas de bases de datos. MANEJO EFICIENTE DEL TIEMPO*. 2005.
64. Momjian, B., *PostgreSQL: introduction and concepts*. Vol. 192. 2001: Addison-Wesley.
65. Owens, M., *The definitive guide to SQLite*. 2006: Apress.
66. MySQL, A., *MySQL*. 2004, MySQL Press.
67. Widenius, M., D. Axmark, and P. DuBois, *MySQL reference manual*. 2002: O'Reilly & Associates, Inc.
68. Le Yang, Y.C., et al., *Design of VY: A Mini Visual IDE for the Development of GUI in Embedded Devices*. 2007.
69. i Noguerras, G.B., *Introducción informal a Matlab y Octave*. 2005.
70. "Open Source Definition, v1.9". Open Source Initiative. 2005.

72. Campaña “Nosotros Hablamos de Software Libre”. Free Software Foundation Europa. 2005.
73. “Categorías de Software Libre y No Libre”. Free Software Foundation. 1.998.
74. Morales, J. L. A. (2006). El software privativo. Retrieved from <http://jalvarezm.wordpress.com/2006/09/30/el-software-privativo/>

BIBLIOGRAFÍA

1. Hortas, M.O., *Introducción a la Biometría*. 2006.
2. BIOMETRICS-ON (2008) *Historia sobre la biometría*.
3. Jain, A.K., P.J. Flynn, and A.A. Ross, *Handbook of biometrics*. 2008: Springer-Verlag New York Inc.
4. BRENNAN, R.G. (2006) *Seguridad vs. Privacidad*.
5. Sartori, R.A.S., *Sistemas de Identificación Biométrica Modernas y Biodinámicas*. 2007.
6. FUENMAYOR, I.G., *AVANCES EN TÉCNICAS BIOMÉTRICAS Y SUS APLICACIONES EN SEGURIDAD*. 2005.
7. Maltoni, D., *Handbook of fingerprint recognition*. 2003: Springer-Verlag New York Inc.
8. Cerón, R.A., *Arquitecturas del Software*.
9. Pressman, R.S., *Ingeniería de software*. 2000.
10. Bastarrica, M.C., *Atributos de Calidad y Arquitectura del Software*. 2005, Volumen.
11. España, S., et al. *Evaluación de la Usabilidad en un Entorno de Arquitecturas Orientadas a Modelos*. 2006.
12. Mezo, B.M., T.S. Chaparro, and A.D. Heras, *Características de las empresas que utilizan Arquitectura Orientada a Servicios y de su contexto de operación*. Journal of Information Systems and Technology Management, 2008. 5(2): p. 269-304.
13. Abascal, J. and R. Moriyón, *Tendencias en Interacción Persona Computador*. Inteligencia artificial. Revista iberoamericana de inteligencia artificial. 2002(16): p. 9–24.
14. Cormen, T.H., *Introduction to algorithms*. 2001: The MIT press.
15. Wu, C., *Advanced feature extraction algorithms for automatic fingerprint recognition systems*. . 2007.
16. Fons, M., et al., *Codiseño Hardware-software de un Algoritmo de Matching Biométrico*. 2003, JCRA.
17. Jiménez, Á. and M. Antonia, *Evaluación biométrica de algoritmos de autenticación de firma manuscrita*. 2010.
18. Areitio, J. and T. Areitio, *Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación*.
19. Hugo, G.O.V., *SISTEMA DE RECONOCIMIENTO DE HUELLAS DACTILARES PARA EL CONTROL DE ACCESO A RECINTOS*.
20. Heidi Alina Nuevo Leon, A.M.R.C., Enrique Reyes Bermudez, Claudia Maria Lora Pomar, Yisel Delgado Mesa (2010) *ALGORITMO DE DETECCIÓN FACIAL PARA SISTEMAS DE AUTENTICACIÓN BIOMÉTRICA*. .
21. Grimaldi, D.S. (2005) *RECONOCIMIENTO BIOMÉTRICO. RECONOCIMIENTO FACIAL*.
22. Traslaviña, C.M.G. (2006) *Introducción a la biometría*.
23. NeuroTechnology (2009) *Iris identification for PC and Web solutions, VeriEye SDK*.
24. Novara, P., *Manual Pselnt*. 2006.
25. Ben Fry, C.R., *INTRODUCCIÓN A PROCESSING*. 2001.
26. Tecnológico, O., *JAVA - Entorno de desarrollo jGrasp*. 2009.
27. Kölling, M., *Introducción a la programación con Greenfoot* 2001.
28. Villanueva., C., *Alice: Aprende a Programar con objetos*. 2008.
29. Rodríguez, M.G., *Introducción rápida a Matlab y Simulink para ciencia e ingeniería*. 2003: Ediciones Diaz De Santos SA.
30. Probst, G., D. de Melo, and M. Roloff. *OpenVisionBot—Pesquisa e desenvolvimento de urna plataforma robótica móvel aberta*: IEEE.
31. Alfaro, V.M., *La simulación digital como herramienta en la docencia y la investigación*. Ingeniería, 2000. 10(1): p. 2.
32. Booch, G. and A.W. Brown, *Collaborative development environments*. Advances in Computers, 2003. 59: p. 1-27.
33. Rice, E.H., *The Collaboration Process in Professional Development Schools: Results of a Meta-Ethnography, 1990-1998*. Journal of Teacher Education, 2002. 53(1): p. 55-67.
34. Lebrún, C.A.V., *INTEGRACIÓN DE HERRAMIENTAS DE TECNOLOGÍAS DE INFORMACIÓN "PORTALES COLABORATIVOS DE TRABAJO" COMO SOPORTE EN LA ADMINISTRACIÓN DEL CONOCIMIENTO*. 2007.
35. Grau, A., *Herramientas de gestión del conocimiento*. Disponible en: www.gestiondelconocimiento.com/documentos2/america/herramientas.htm [Consultado: 10 de noviembre del 2004].
36. Gaudioso Vázquez, E., et al., *Cómo gestionar la colaboración en el Marco Lógico Colaborativo en un entorno de aprendizaje adaptativo basado en web*. Inteligencia artificial, 2004. 8(24): p. 121-129.

37. Churchill, E.F., D.N. Snowdon, and A.J. Munro, *Collaborative virtual environments: digital places and spaces for interaction*. Educational Technology & Society, 2002. **5**(4).
38. Accino, J.A., *ÁGORA VIRTUAL: una propuesta de entorno colaborativo y de enseñanza sobre interfaces OSID*. Boletín RedIRIS, 2006.
39. Maybury, M., *Collaborative virtual environments for analysis and decision support*. Communications of the ACM, 2001. **44**(12): p. 51-54.
40. Simón, A., et al. *GECOSOFT: Un Entorno Colaborativo para la Gestión del Conocimiento con Mapas Conceptuales*. 2006.
41. GRZEGA, J., *la Wikipedia*. Onomasiology Online, 2006. **7**: p. 1-15.
42. Gomez, M.V., *Wikipedia: la enciclopedia libre*.
43. Broncano, R. (2009) *Analizamos Feng Office, ¿la oficina del futuro?*
44. Leo, J. (2010) *Doolphy, otra plataforma para gestionar proyectos y tareas en grupo*.
45. Fernández, A., *Redmine, gestión de proyectos web*. Todo linux, 2010(120): p. 32-35.
46. López, J., *Redmine, gestión de proyectos Ruby on Rails*. Todo linux, 2009(104): p. 21.
47. Tramullas, J., *Herramientas de software libre para la gestión de contenidos*. Hipertext. net, 2005. **3**.
48. Unblogenred.es, *El Proyecto Vulcano potencia el software libre de calidad en Europa*. 2010.
49. vulcano (2007) *Vulcano*.
50. Copeland, T., et al., *GForge Manual, 2006*. URL http://gforge.org/docman/view.php/1/34/gforge_manual.pdf.
51. Red, U., *Media Art Wiki. Uso de Wikis para la enseñanza interdisciplinar y multimedia del arte de los nuevos medios de comunicación en entornos virtuales de aprendizaje*. Revista de Docencia Universitaria, 2010(0).
52. González, M. (2010) *La mediación artística: un modelo de educación artística para la intervención social a través del arte*.
53. <http://issuu.com/> (2009) *Poliedro On Line*.
54. Luna, F.E. (2009) *Poliedro [On line]*
55. <http://narratopedia.net/>. *Narratopedia*. 2009; Available from: <http://narratopedia.net/>.
56. Calderón, J.D. and A. de Servidores, *CAPACITACION DE SISTEMA DE CONTROL DE VERSIONES BAZAAR*.
57. PROUDHON, P.J., *Concurrent Versions System*. Software Studies: p. 64.
58. Collins-Sussman, B., B.W. Fitzpatrick, and C.M. Pilato, *Control de versiones con Subversion*. 2002.
59. García Domínguez, A., *Introducción al Sistema de Control de Versiones Distribuido Git*. 2011.
60. López, J., *Mercurial, el sistema de control de versiones distribuido*. Todo Linux: la revista mensual para entusiastas de GNU/Linux, 2009. **9**(104): p. 46-49.
61. Connolly, T.M.a.C.E.B., *Sistemas de bases de datos. MANEJO EFICIENTE DEL TIEMPO*. 2005.
62. Momjian, B., *PostgreSQL: introduction and concepts*. Vol. 192. 2001: Addison-Wesley.
63. Owens, M., *The definitive guide to SQLite*. 2006: Apress.
64. Widenius, M., D. Axmark, and P. DuBois, *MySQL reference manual*. 2002: O'Reilly & Associates, Inc.
65. MySQL, A., *MySQL*. 2004, MySQL Press.
66. Le Yang, Y.C., et al., *Design of VY: A Mini Visual IDE for the Development of GUI in Embedded Devices*. 2007.
67. i Nogueras, G.B., *Introducción informal a Matlab y Octave*. 2005.

GLOSARIO DE TÉRMINOS.

SDK: Conjunto de herramientas de desarrollo que le permite a un programador crear aplicaciones para un sistema concreto.

GPU: Una GPU es un chip dedicado exclusivamente al procesamiento de gráficos. Las GPU son muy potentes y pueden incluso superar la frecuencia de reloj de una CPU antigua (más de 500MHz).

Cross Match I Scan 2 y Retica Mobile-Eyes: Son una parte integral de cualquier programa completo para la administración de identidad biométrica. El dispositivo provee una inscripción móvil rápida, identificación y verificación a través de sus aplicaciones para la captura de iris a cualquier instalación de seguridad.

C++: Es un lenguaje de programación diseñado a mediados de los años 1980 por Bjarne Stroustrup. La intención de su creación fue el extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Licencia GPL: Permite la copia, modificación y redistribución del software. Al no suponer costo alguno, tampoco nos ofrece garantías. Posibilita que el software sea vendido y se puedan cobrar por los servicios sobre las aplicaciones creadas.

MIDI: (Interfaz Digital de Instrumentos Musicales). Se trata de un protocolo de comunicación serial estándar que permite a los computadores, sintetizadores, secuenciadores, controladores y otros dispositivos musicales electrónicos comunicarse y compartir información para la generación de sonidos.

XML: El lenguaje que nos garantizará el intercambio de cualquier tipo de información, sin que ocasione problemas de tipo "contenido" o de tipo "presentación". Este garantiza que los datos estructurados sean uniformes e independientes de aplicaciones o fabricantes, lo que está originando una nueva generación de aplicaciones en la Web.

Bookmarks: Es la localización almacenada de una página web de forma que puede ser revisitada más adelante. Los enlaces guardados se suelen denominar favoritos en Internet Explorer y marcadores en Mozilla Firefox, los dos navegadores web más utilizados en el mercado.

3D: El término gráficos 3D por computadora o por ordenador (en inglés 3D computer graphics) se refiere a trabajos de arte gráfico que son creados con ayuda de computadoras y programas especiales 3D. En general, el término puede referirse también al proceso de crear dichos gráficos, o el campo de estudio de técnicas y tecnología relacionadas con los gráficos 3D.

OpenCV: Es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Es multiplataforma, Existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica.

Buzzword: Es un procesador de textos colaborativo en línea, sencillos y comodísimo de usar. Posee un interfaz sencillo pero elegante, con pocas opciones, las imprescindibles de un procesador de texto en línea, soportando imágenes, listas, estilos, tablas, comentarios en el documento. Permite editar por varios usuarios a la vez e incluye un diccionario personalizable.

Google Docs: Es un programa gratuito basado en Web para crear documentos en línea con la posibilidad de colaborar en grupo. Incluye un Procesador de textos, una Hoja de cálculo, Programa de presentación básico y un editor de formularios destinados a encuestas.

Etherpad: Es una sencilla aplicación web que permite editar el mismo documento (Pad) de manera simultánea por varias personas en tiempo real. Tiene gran sencillez de uso, hasta el punto de que en su versión básica no hace falta ni tener usuario para utilizarla.

Wiki: Un wiki, o una wiki, es un sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web. Los usuarios pueden crear, modificar o borrar un mismo texto que comparten.

Blog: Es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

Moodle: Es un Ambiente Educativo Virtual, sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Servicio web: (en inglés, *Web service*) Es una pieza de software que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Protocolo HTTP: El protocolo de transferencia de hipertexto (HTTP, Hyper Text Transfer Protocol (Protocolo de transferencia de Hyper Texto)) es el protocolo usado en cada transacción de la Web (WWW). Un protocolo se puede entender como el lenguaje utilizado por dos computadoras para comunicarse entre sí.

PHP: Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.

Framework: Estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Diagramas de Gantt: Es una popular herramienta gráfica cuyo objetivo es mostrar el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

Diagrama PERT: Es una representación gráfica de las relaciones entre las tareas del proyecto que permite calcular los tiempos del proyecto de forma sencilla.

MS Project: Es un software de administración de proyectos diseñado, desarrollado y comercializado por Microsoft para asistir a administradores de proyectos en el desarrollo de planes, asignación de recursos a tareas, dar seguimiento al progreso, administrar presupuesto y analizar cargas de trabajo.

Plataforma INES: Es la Plataforma Tecnológica Española en el área los Sistemas y Servicios Software y constituye una red de cooperación científico-tecnológica integrada por los agentes tecnológicos relevantes del área (empresas, universidades, centros tecnológicos, entre otros).

Plug-ins: Programas que deben instalarse en los ordenadores clientes para que sus navegadores puedan reconocer y procesar determinados tipos de archivo (Flash, Director, Real Audio).

GNU C: Es un conjunto de compiladores creados por el proyecto GNU (sistema operativo completamente libre).

Apache: Es un servidor web HTTP de código abierto para plataformas Unix.

FTP: (sigla en inglés de File Transfer Protocol - Protocolo de Transferencia de Archivos) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor.

Mac OS X: es un sistema operativo desarrollado y comercializado por Apple Inc., el cual ha sido incluido en los sistemas Macintosh desde 2002.

CIDR: representa la última mejora en el modo como se interpretan las direcciones IP. Su introducción permitió una mayor flexibilidad al dividir rangos de direcciones IP en redes separadas.

Direcciones MAC: (siglas en inglés de Media Access Control o control de acceso al medio) es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una ethernet de red.

ACID: conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

ANEXOS

Anexo 1. Guía para la evaluación de la propuesta.

Fecha de recepción _____

Fecha de entrega _____

Experto # _____

Usted le otorgará un peso a cada criterio de acuerdo a su opinión teniendo en cuenta la siguiente escala de puntuación:

Baja: 6 o menos

Media: 7 u 8

Alta: 9 ó 10

Criterios de medida que se evalúan en una escala de 1 – 10

Grupo No. 1: Criterios de mérito científico

1. Valor científico de la propuesta.

Peso: _____

2. Calidad de la investigación.

Peso: _____

Grupo No. 2: Criterios implantación

3. Necesidad del empleo de la propuesta.

Peso: _____

4. Posibilidades de aplicación.

Peso: _____

5. Satisfacción de las necesidades de la producción.

Peso: _____

Grupo No.3: Criterios de flexibilidad

6. Adaptabilidad al Entorno Colaborativo de otros proyectos productivos del CISED.

Peso: _____

7. Eficiencia y calidad de la propuesta del Entorno Colaborativo para el Desarrollo de Algoritmos Biométricos.

Peso: _____

Grupo No.4: Criterios de impacto

8. Repercusión del Entorno propuesto para desarrollar algoritmos biométricos.

Peso: _____

9. Aceptación de la propuesta por los directivos del Departamento.

Peso: _____

10. Impacto de la propuesta en el área para la cual está destinada.

Peso: _____

Grupo No 5: Criterios de generalización.

11. La propuesta es de fácil entendimiento.

Peso: _____

12. No es necesario ser un especialista en el desarrollo de algoritmos biométricos para ponerla en práctica.

Peso: _____

Categoría final del proyecto

___ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

___ Bueno: Novedad científica, resultados destacados.

___ Aceptable: Suficientemente bueno con reservas.

___ Cuestionable: No tiene relevancia científica y los resultados son malos.

___ Malo: No aplicable.

Sugerencias del evaluador para mejorar la calidad de la propuesta.

Elementos críticos que deben mejorarse: _____.