

**UNIVERSIDAD NACIONAL DE SAN CRISTÓBAL DE HUAMANGA**  
**FACULTAD DE INGENIERÍA DE MINAS GEOLOGÍA Y CIVIL**  
**ESCUELA DE FORMACIÓN PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**“SOFTWARE DE CONTROL BIOMÉTRICO MEDIANTE HUELLA DIGITAL, PARA VERIFICAR LA IDENTIDAD DE UN ACTOR DE LA UNSCH, 2016”**

**Tesis presentada por** : Bach. Juan Antonio Gabino Candia

**Para optar el título de** : Ingeniero de Sistemas

**Tipo de investigación** : Descriptiva y aplicada

**Área de investigación** : Ingeniería de Software

**Asesor** : MSc. Ing. Efraín Elías Porras Flores

**AYACUCHO – PERÚ**

**2016**

## **AGRADECIMIENTO**

A la Universidad Nacional de San Cristóbal de Huamanga, mi alma mater y a todos los docentes de la universidad.

A mis padres, por su apoyo y motivación constante, pero sobre todo, por el amor incondicional que me brindaron en todo momento.

A mis hermanos, por su apoyo incondicional, ya que a pesar de las dificultades, nos mantenemos siempre unidos.

## **DEDICATORIA**

A mi madre Eduarda (QEPD), quien me brindó su cariño y apoyo incondicional para seguir adelante a pesar de las dificultades.

A mi padre Antonio por su constante apoyo y sacrificio para mi desarrollo profesional.

A mis hermanos Rosa y Percy por brindarme su alegría, y apoyo en todo momento.

A mis maestros, quienes supieron guiarme en esta etapa, contribuyendo a mi formación profesional, brindándome sus conocimientos y enseñanzas.

# CONTENIDO

AGRADECIMIENTO .....	i
DEDICATORIA .....	ii
CONTENIDO .....	iii
RESUMEN.....	v
INTRODUCCIÓN .....	vi

## CAPÍTULO I

### PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1 DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA .....	1
1.2 DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN .....	2
1.3 DELIMITACIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN .....	2
1.4 JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN .....	3

## CAPÍTULO II

### REVISIÓN LITERARIA

2.1 ANTECEDENTES .....	5
2.2 MARCO TEÓRICO .....	7
2.2.1 PROCESOS CRÍTICOS DE GESTIÓN UNIVERSITARIA .....	7
2.2.1.1 GESTIÓN ACADÉMICA.....	9
2.2.1.2 GESTIÓN ADMINISTRATIVA .....	10
2.2.2 ACTORES .....	11
2.2.2.1 ESTUDIANTE .....	12
2.2.2.2 DOCENTE UNIVERSITARIO .....	13
2.2.2.3 POSTULANTE .....	14
2.2.3 HUELLA DIGITAL .....	14
2.2.3.1 RENDIMIENTO BIOMÉTRICO .....	16
2.2.4 LECTOR DE HUELLA DIGITAL .....	18
2.2.5 METODOLOGÍA DE DESARROLLO ÁGIL DE PROGRAMACIÓN EXTREMA .....	20
2.2.6 SISTEMA GESTOR DE BASE DE DATOS RELACIONAL .....	30
2.2.7 LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS.....	35
2.2.7.1 JAVA .....	41
2.2.8 KIT DE DESARROLLO DE SOFTWARE (SDK) .....	44
2.2.8.1 GRIAULE FINGERPRINT SDK .....	44
2.2.9 POBLACIÓN Y MUESTRA .....	45
2.2.10 MUESTREO.....	47

### **CAPÍTULO III**

#### **METODOLOGÍA DE LA INVESTIGACIÓN**

3.1	TIPO DE INVESTIGACIÓN .....	50
3.2	NIVEL DE INVESTIGACIÓN.....	50
3.3	DISEÑO DE LA INVESTIGACIÓN .....	51
3.4	POBLACIÓN Y MUESTRA .....	51
3.5	VARIABLES E INDICADORES .....	51
3.6	TÉCNICAS E INSTRUMENTOS PARA EL LEVANTAMIENTO DE INFORMACIÓN .....	53
3.6.1	HERRAMIENTAS PARA EL TRATAMIENTO DE LA INFORMACIÓN.....	54
3.6.2	TÉCNICAS PARA APLICAR LA METODOLOGÍA XP .....	55

### **CAPÍTULO IV**

#### **ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN**

4.1.	ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP .....	57
4.1.1	FASE DE EXPLORACIÓN .....	57
4.1.2	FASE DE PLANIFICACIÓN.....	58
4.1.3	FASE DE ITERACIÓN.....	63
4.1.4	PLAN DE ITERACIÓN .....	70
4.2.	INTERFAZ DE USUARIO Y CODIFICACIÓN.....	72
4.3.	TARJETAS CLASE RESPONSABILIDAD Y COLABORACIÓN (CRC) .....	75
4.4.	IMPLEMENTACIÓN.....	79
4.5.	CASOS DE PRUEBA Y ACEPTACIÓN.....	81

### **CAPÍTULO V**

#### **CONCLUSIONES Y RECOMENDACIONES**

5.1.	CONCLUSIONES.....	88
5.2.	RECOMENDACIONES.....	89

BIBLIOGRAFÍA.....	90
-------------------	----

ANEXOS .....	93
--------------	----

## RESUMEN

La Universidad Nacional de San Cristóbal de Huamanga, es una institución educativa pública, que cuenta con múltiples procesos, entre los cuales existen procesos críticos de gestión universitaria, la ejecución de estos procesos no garantiza la identidad de los actores que los ejecutan.

El presente estudio tiene como objetivo automatizar los procesos críticos de gestión universitaria, aplicando el control por huella digital, para garantizar la identidad de los actores de la Universidad Nacional de San Cristóbal de Huamanga, a través de un software de control biométrico basado en un lector de huella digital. La investigación se llevó a cabo en la ciudad de Ayacucho y es una investigación de tipo descriptiva, nivel aplicada, de diseño no experimental transeccional. Para el desarrollo del software se usó el lenguaje de programación orientado a objetos y el proceso ágil de programación extrema. Para la recolección de datos se usó las técnicas de análisis documental, entrevistas y cuestionarios.

De acuerdo al capítulo IV se obtuvo los artefactos de historias de usuario definidas en la Tabla N° 4.1 de la sección 4.1.1, las historias de usuarios desarrolladas en las Tablas N° 4.3, 4.4, 4.5, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12, el modelo físico de base de datos en la figura N° 4.11 y el diagrama de clases en la figura N° 4.12, usando la metodología de desarrollo ágil de programación extrema. Se logró verificar la identidad de los estudiantes, docentes universitarios y postulantes.

**PALABRAS CLAVES:** Biometría, huella digital, actor de la universidad, procesos críticos de gestión universitaria.

## **INTRODUCCIÓN**

Las huellas digitales son únicas en cada ser humano, con este razonamiento y sus diferentes características las cuales permiten que puedan ser clasificadas, el ser humano las está usando para poder identificar a las personas, y es uno de los métodos de reconocimiento más confiables hoy en día (Carrión, 2009). Los sistemas de seguridad para el acceso a lugares, verificación o identificación de personal, basados en utilizar rasgos biométricos se presentan como el futuro en el campo de la seguridad puesto que son los más confiables y actualmente están siendo usados en muchas partes del mundo sin mayores problemas. (Balmelli, 2006). La programación extrema (XP) es una metodología de desarrollo ligera o ágil basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas (Aguilar, 2002).

Lo que se busca con la presente investigación, es resolver el problema de automatización de los procesos críticos de gestión universitaria, garantizando la seguridad al ejecutar dichos procesos críticos por los actores de la universidad. El software de control biométrico por huella digital, permite que los actores sean identificados de una manera rápida y fiable, antes de ejecutar un proceso crítico, garantizando así la identidad de dichos actores.

Actualmente el problema de seguridad en la ejecución de los procesos críticos de gestión universitaria, persiste debido a que los procesos se ejecutan de manera manual, generando demora en los procesos y no se garantiza que los procesos sean ejecutados por los actores que dicen ser.

Los objetivos específicos son explorar, planificar e iterar la automatización de los procesos críticos de gestión universitaria, para verificar la identidad tanto de

los estudiantes, usuarios de biblioteca, comensales, usuarios del autoseguro, docentes universitarios y postulantes, con la finalidad de garantizar que los procesos críticos de gestión universitaria sean ejecutados por los actores que dicen ser.

# **CAPÍTULO I**

## **PLANTEAMIENTO DE LA INVESTIGACIÓN**

### **1.1 DIAGNÓSTICO Y ENUNCIADO DEL PROBLEMA**

La Universidad Nacional de San Cristóbal de Huamanga, es una institución educativa pública que tiene como fin, la formación académica de profesionales competitivos en distintas especialidades, y para cumplirlo se realizan múltiples procesos, en los cuales están involucrados 3 tipos de actores; los estudiantes, postulantes y docentes universitarios. Dentro de los múltiples procesos que se realizan, se puede encontrar procesos críticos, tales como; controlar el acceso de los postulantes al examen de Admisión de la Universidad y controlar la asistencia de docentes a sus labores diarias, entre otros procesos, de los cuales no se está garantizando la identidad del actor.

Actualmente, el proceso para controlar los postulantes al examen de admisión, se realiza de forma manual, lo que ocasiona que al momento de controlar el acceso al examen de admisión, se generen largas colas en las inmediaciones de la universidad, y así demore el acceso a los ambientes, además que no se garantiza la identidad de los postulantes, aumentando así la probabilidad que ocurra algún tipo de suplantación de postulantes al examen de admisión. Luego del examen de admisión, se realiza el proceso de entrega de constancias de ingreso, en el cual los postulantes que aprobaron el examen y ocuparon las vacantes disponibles, reciben una constancia, que acredita el vínculo del estudiante con la Universidad. En este proceso, la identificación del estudiante se hace por medio del documento de identidad, generando que exista la posibilidad que, la persona que rindió el examen de admisión, no sea la persona a quien se le entregue la constancia de ingreso.

Otro problema que se viene presentando en la universidad, es que no se realiza

un control de adecuado de la asistencia de los docentes al dictado de sus clases, además que no se garantiza que los documentos de gestión sean emitidos por los docentes que corresponden, y que estos documentos a su vez sean emitidos en el tiempo establecido.

## **1.2 DEFINICIÓN DEL PROBLEMA DE INVESTIGACIÓN**

### **PROBLEMA PRINCIPAL**

¿De qué manera automatizar los procesos críticos de gestión universitaria, para garantizar la identidad de los actores de la Universidad Nacional de San Cristóbal de Huamanga, 2016?.

### **PROBLEMAS SECUNDARIOS**

- a) ¿De qué manera automatizar los procesos críticos de gestión universitaria, para verificar la identidad de los estudiantes?.
- b) ¿De qué manera automatizar los procesos críticos de gestión universitaria, para verificar la identidad de los docentes?.
- c) ¿De qué manera automatizar los procesos críticos de gestión universitaria, para verificar la identidad de los postulantes?.

## **1.3 DELIMITACIÓN DE LOS OBJETIVOS DE LA INVESTIGACIÓN**

### **OBJETIVO GENERAL**

Desarrollar un software para automatizar procesos críticos de gestión universitaria mediante técnicas e instrumentos, la metodología de desarrollo ágil de programación extrema, un sistema gestor de base de datos relacional, un lenguaje de programación orientado a objetos y un lector de huella digital, con la finalidad de garantizar la identidad de los actores de la Universidad Nacional de San Cristóbal de Huamanga, 2016.

### **OBJETIVOS ESPECÍFICOS**

- a) Explorar, planificar e iterar la automatización de los procesos críticos de gestión universitaria, con la finalidad de verificar la identidad de los estudiantes, usuarios de biblioteca, comensales y usuarios del

- autoseguro.
- b) Explorar, planificar e iterar la automatización de los procesos críticos de gestión universitaria, con la finalidad de verificar la identidad de los docentes universitarios y garantizar la autenticidad de los documentos de gestión emitidos.
  - c) Explorar, planificar e iterar la automatización de los procesos críticos de gestión universitaria, con el fin de verificar la identidad de los postulantes.

#### **1.4 JUSTIFICACIÓN Y DELIMITACIÓN DE LA INVESTIGACIÓN IMPORTANCIA Y JUSTIFICACIÓN**

La presente investigación se llevó a cabo para automatizar los procesos críticos relacionados a la identidad de actores de la Universidad Nacional de San Cristóbal de Huamanga, debido que actualmente, no se garantiza totalmente la identidad de los actores.

En el aspecto social, al automatizar los procesos críticos aplicando un control por huella digital, se logrará que los procesos que se realicen, sean transparentes y no genere especulaciones sobre que se haya cometido algún tipo de fraude en algunos procesos críticos, como es el proceso del examen de admisión, en el que ocurren casos de suplantación de postulantes; además, se hará un mejor control de asistencia de los docentes, y cada proceso crítico, se ejecutará por el actor que corresponde.

En el aspecto económico, el software reducirá gastos innecesarios como los que ocurren para el control de acceso de los postulantes al examen de admisión, al contratar demasiadas personas para identificar a un solo postulante, además que en la actualidad, el precio de los dispositivos biométricos de huella digital, son relativamente económicos en comparación a los marcadores de asistencia por huella digital, que se vienen usando actualmente para controlar al personal administrativo.

En el aspecto técnico se utilizó el sistema de control por huella digital, para controlar el acceso de postulantes, validar la identidad de los ingresantes, controlar la asistencia de los docentes, entre otros procesos. Los lectores de huella digital, en la actualidad vienen mejorando progresivamente, y aumentando su grado de eficiencia, lo cual conlleva a que este tipo de sistemas nos garantice la identidad de personas a través de la huella digital de cada persona.

### **DELIMITACIÓN**

La investigación se realizó en la ciudad de Ayacucho, con los actores que participan en los procesos críticos de la Universidad Nacional San Cristóbal de Huamanga, en el año 2016.

## **CAPÍTULO II**

### **REVISIÓN LITERARIA**

#### **2.1 ANTECEDENTES**

Como afirma Balmelli (2006), los sistemas de seguridad para el acceso a lugares, verificación o identificación de personal, basados en utilizar rasgos biométricos se presentan como el futuro en el campo de la seguridad puesto que son los más confiables y actualmente están siendo usados en muchas partes del mundo sin mayores problemas.

En el Perú, como en la gran mayoría de países en vías de desarrollo, la falta de tecnología es algo común por lo que el uso de sistemas biométricos para la seguridad sería un gran avance. Teniendo en cuenta que la implementación de sistemas biométricos no necesariamente sería algo costoso, puesto que hay lectores biométricos con costos bajos ya que esta tecnología va avanzando y progresando rápidamente, habría que tomar en cuenta la idea de ir mejorando y creciendo junto con la tecnología.

La implementación de estos sistemas significaría un mayor control de acceso físico, evitaría largos tiempos de espera para ingresar a ciertos lugares públicos y/o privados como sería el caso a una universidad por parte de los alumnos, protegería transacciones financieras, verificaría el tiempo de llegada y salida de empleados de sus centros laborales, evitaría robos y plagios, siendo estos últimos ejemplos dos de las mayores razones para la utilización de sistemas de seguridad más confiables.

Como afirma Simón (2003), los métodos tradicionales de autenticación presentan el inconveniente de que no pueden discriminar de manera fiable entre los individuos legítimos y los individuos impostores; ya que la identidad

que la persona tiene puede ser sustraída, perdida, etc., y la entidad que la persona sabe puede ser olvidada o confundida. En cambio, los métodos basados en la autenticación de la identidad por medio de los rasgos biométricos de un individuo proporcionan una mayor fiabilidad en la identificación personal, ya que no pierden, no se olvida, ni tampoco se puede compartir. En combinación con los métodos de autenticación por posesión y/o conocimiento permiten configurar sistemas de identificación personal muy confiables.

Como afirma Carrión (2009), las huellas digitales son únicas en cada ser humano, con este razonamiento y sus diferentes características las cuales permiten que puedan ser clasificadas, el ser humano las está usando para poder identificar a las personas es uno de los métodos de reconocimiento más confiables hoy en día. En nuestra realidad, la Dirección de Criminalística (DIRCRI) de la Policía Nacional del Perú usa las impresiones de las huellas las cuales son clasificadas según la forma de las crestas papilares características de cada individuo. Dado que en la actualidad la clasificación requiere de un proceso visual y el almacenamiento en tarjetas dactiloscópicas genera que estas no tengan un control de cantidad se busca que el proceso de clasificación lo haga más rápido y eficiente.

Como afirma López (s.f.), las primeras aplicaciones de las técnicas biométricas tuvieron lugar dentro del ámbito legal, particularmente en el campo forense. Sin embargo, en las últimas dos décadas, la expansión tecnológica en nuestra sociedad, ha creado la necesidad de diseñar sistemas automáticos de alta seguridad capaces de identificar a los diferentes individuos a partir de sus rasgos biométricos. Cada uno de estos rasgos presenta sus ventajas e inconvenientes dependiendo del escenario en el que se desarrollan las aplicaciones y del grado de fiabilidad requerida. El método de identificación mediante huella digital es uno de los métodos más fiables que actualmente se conocen y en pocos años posiblemente se incorporará en nuestra vida diaria. En estos últimos años la biometría por huella digital se ha acercado al público en general y casi no nos resulta extraño ver en algunas instalaciones la

utilización de detectores de huella digital para el acceso de personas, incluso en los ordenadores portátiles incluyen detectores de huella digital para que un usuario previamente registrado pueda iniciar una sesión.

“Las soluciones tecnológicas, como lo son las biométricas (huella digital), nos permiten mediante la ayuda de un dispositivo lector de huellas digitales, comparar las huellas de la persona que se está identificando frente a una base de datos de huellas de personas que cuenten con los permisos necesarios para realizar determinada labor o para ingresar a determinado recinto. Su trabajo de tesis se centra, en el estudio de cómo una organización puede identificar a las personas que ingresan a sus recintos haciéndoles un seguimiento y obtener información acerca de la ocurrencia del ingreso de las personas a la organización: hora de ingreso y salida, tiempo de permanencia en la organización, labores realizadas, etc.” Cernandes y Zapata (2006).

## **2.2 MARCO TEÓRICO**

### **2.2.1 PROCESOS CRÍTICOS DE GESTIÓN UNIVERSITARIA**

“Proceso crítico, es un proceso que se define como vital para el funcionamiento del negocio, a veces durante un período. La definición surge de los criterios empleados en la práctica Priorizar.” (Bravo, 2015).

Como afirma Sarracino (s.f.), un proceso es crítico cuando en gran medida la consecución de los objetivos y los niveles de calidad de la empresa dependen de su desarrollo.

Por gestión universitaria, entendemos la “capacidad de generar las mejores condiciones para que los procesos institucionales ocurran con eficiencia y eficacia, en la consecución de objetivos y metas, mediante una relación adecuada entre la estructura, las estrategias, los liderazgos y las capacidades de los recursos humanos disponibles.” (Padilla, 2006).

La gestión universitaria puede ser entendida como el conjunto de estrategias dirigidas por personas, cuerpos colegiados y de mando directivo, para garantizar el cumplimiento del proyecto educativo bajo condiciones sostenibles y viables. (Lopera, 2004).

“La gestión universitaria implica una serie de acciones, decisiones y políticas que lleva a cabo la autoridad institucional sobre los procesos académicos y administrativos, mismos que están orientados a que las instituciones educativas cumplan a cabalidad las funciones sustantivas para lo que fueron creadas. Es decir, estas acciones, decisiones y políticas tienen como finalidad central el que las instituciones educativas instrumenten actividades administrativas de docencia, de investigación, de difusión de la cultura para el mejoramiento de la eficiencia y la eficacia de los sistemas educativos.” (Valdez et al., 2011).

Las acciones más importantes que realiza la gestión universitaria y que directa o indirectamente inciden en los procesos de seguimiento y mejora continua son:

- a) Aprobación y actualización de los planes de estudios.
- b) Capacitación y actualización disciplinar y pedagógica de la planta docente.
- c) Control de asistencia de los docentes y alumnos. Una acción específica, como parte de las actividades adjetivas de las instituciones de educación superior y que forma parte de la gestión universitaria, es el control de asistencia de los docentes y éstos, a su vez, de los alumnos. Si el alumno asiste a clases hay garantía de que el proceso de aprendizaje puede realizarse, sea este con deficiencias o no, pero si no asiste a clases la probabilidad de que este proceso se realice es muy baja. De ahí, la importancia de la gestión universitaria para garantizar condiciones básicas para la realización del proceso de aprendizaje.
- d) Establecimiento de políticas y modelos académicos centradas en el aprendizaje.
- e) Incorporación de nuevos profesores.
- f) Gestión de las políticas de internacionalización de la educación.

- g) Planeación e instrumentación de la evaluación del desempeño docente y retroalimentación. La evaluación de la educación es una de las estrategias de mayor incidencia en la mejora, el desarrollo y la innovación de la educación superior.
- h) Coordinación e implementación de los programas de intervención tutorial.
- i) Facilitadores de las nuevas tecnologías en apoyo al proceso de enseñanza aprendizaje.
- j) Programas remediales para alumnos reprobados y con problemas de aprendizaje.
- k) Programas de mantenimiento, acondicionamiento y mejora de la infraestructura educativa.
- l) Creación y coordinación del programa de extensión y vinculación universitaria.
- m) Coordinación de acreditación de los programas de estudio y certificación de los docentes.

### **2.2.1.1 GESTIÓN ACADÉMICA**

La gestión académica es un proceso complejo que involucra la entrada de recursos diversos (tangibles e intangibles), un procesamiento de la complejidad más elevada que pueda existir (pues tiene que vérselas con el desarrollo de las capacidades intelectuales y emotivas, que involucra aspectos aptitudinales y actitudinales), y genera salidas bajo la forma de productos de alta complejidad (como: nuevos conocimientos, profesionalidad, habilidades cognoscitivas, investigativas, capacidades de solución en el descubrimiento, formulación, planteamiento y resolución de problemas profesionales, pretendiendo que se minimicen los errores y se maximicen los aciertos en aras de garantizar el continuado progreso de la sociedad humana en equilibrada armonía con la naturaleza a la que pertenece).

El proceso de gestión académica que realizan los departamentos docentes suelen presentar distinta complejidad según las tareas que le correspondan. Una de las distinciones más significativas está asociada al hecho de que tengan

la responsabilidad de una carrera o especialidad o el que no tengan como contenido el trabajo en esta dirección. El caso que aquí se estudia es, precisamente, el de una estructura conocida como Departamento-carrera. Ello aporta complejidades adicionales al trabajo académico, tanto en su conducción como en su previsión, orientación, control, medición y evaluación de los resultados.

En el orden de la gestión académica la primera necesidad que se evidencia es la de un enfoque integrador de la diversidad y complejidad de este tipo de actividad, lo que requiere de un sistema de administración que se adecue a esta exigencia objetiva del proceso que se conducirá. La dirección del departamento docente en la educación superior, en tanto administración de la actividad académica, exige que sea considerada en toda su compleja multilateralidad. Ella no está presente en la estructura jerárquica tradicional con que se ha enfocado la administración de la actividad académica, y el sistema de normativas que rigen este trabajo actualmente no lo contemplan de este modo.

#### **2.2.1.2 GESTIÓN ADMINISTRATIVA**

La gestión administrativa es el proceso de diseñar y mantener un entorno en el que, trabajando en grupos, los individuos cumplen eficientemente objetivos específicos. Es un proceso muy particular consistente en las actividades de planeación, organización, ejecución y control, desempeñados para determinar y alcanzar los objetivos señalados con el uso de seres humanos y otros recursos. Existen cuatro elementos importantes que están relacionados con la gestión administrativa; sin ellos es imposible hablar de gestión administrativa, estos son: Planeación Organización, Ejecución, Control. (García, 2008).

De acuerdo al Estatuto Universitario (2015), la gestión administrativa es responsable de conducir los procesos de administración de los recursos humanos, materiales y financieros que garanticen servicios de calidad, equidad y pertinencia.

La Dirección General de Administración tiene las siguientes funciones:

- a. Formular, ejecutar y evaluar el plan anual de trabajo de la Dirección General de Administración, sobre las actividades de su competencia e informar periódicamente al rector.
- b. Ejecutar las fases de ejecución presupuestal del pliego a través del sistema integrado de administración financiera del sector público-SIAF.
- c. Coordinar con la Oficina de Planificación y Presupuesto sobre ejecución de gastos por metas.
- d. Planificar, organizar y dirigir la ejecución de los sistemas y dependencias a su cargo.
- e. Supervisar las funciones de las oficinas adscritas a su dependencia: Oficina de Contabilidad y Tesorería, Oficina de Logística y Control Patrimonial, Oficina de Recursos Humanos, Oficina de Inversión y Servicios Generales, Oficina de Bienestar Universitario, Oficina de Biblioteca, y la Oficina de Informática y Sistemas.
- f. Propone directivas para cada sistema e implementa su aplicación, garantizando la racionalidad y oportunidad de la gestión.
- g. Elaborar, desarrollar y aplicar normas, métodos y procedimientos en relación a las acciones y procesos técnicos de los sistemas.
- h. Resolver por delegación o atribuciones, asuntos administrativos señalados por el Consejo Universitario o el rector.
- i. Proponer al Consejo Universitario, a través del Rectorado, la conformación de comités o comisiones de carácter administrativo.
- j. Atender y resolver los recursos administrativos interpuestos por los usuarios según el texto único de procedimientos administrativos.
- k. Formular y ejecutar programas de capacitación en forma coordinada con las diferentes dependencias de la universidad.
- l. Presentar su memoria anual al Rectorado. m. Cumplir las demás funciones que le asigne el rector.

### **2.2.2 ACTORES**

Según Ceria (s.f.), un actor es una agrupación uniforme de personas,

sistemas o máquinas que interactúan con el sistema que estamos construyendo. Por ejemplo, para una empresa que recibe pedidos en forma telefónica, todos los operadores que reciban pedidos y los ingresen en un sistema de ventas, si pueden hacer las mismas cosas con el sistema, son considerados un único actor: Empleado de Ventas. Los actores son externos al sistema que vamos a desarrollar. Por lo tanto, al identificar actores estamos empezando a delimitar el sistema, y a definir su alcance. Definir el alcance del sistema debe ser el primer objetivo de todo analista, ya que un proyecto sin alcance definido nunca podrá alcanzar sus objetivos.

Es importante tener clara la diferencia entre usuario y actor. Un actor es una clase de rol, mientras que un usuario es una persona que, cuando usa el sistema, asume un rol. De esta forma, un usuario puede acceder al sistema como distintos actores. La forma más simple de entender esto es pensar en perfiles de usuario de un sistema operativo. Una misma persona puede acceder al sistema con distintos perfiles, que le permiten hacer cosas distintas. Los perfiles son en este caso equivalentes a los actores.

Otro sistema que interactúa con el que estamos construyendo también es un actor. Por ejemplo, si nuestro sistema deberá generar asientos contables para ser procesados por el sistema de contabilidad, este último sistema será un actor, que usa los servicios de nuestro sistema.

También puede ocurrir que el actor sea una máquina, en el caso en que el software controle sus movimientos, o sea operado por una máquina. Por ejemplo, si estamos construyendo un sistema para mover el brazo de un robot, el hardware del robot será un actor, asumiendo que dentro de nuestro sistema están las rutinas de bajo nivel que controlan al hardware.

### **2.2.2.1 ESTUDIANTE**

Estudiante, es la palabra que permite referirse a quienes se dedican a la aprehensión, puesta en práctica y lectura de conocimientos sobre alguna

ciencia, disciplina o arte. Es usual que un estudiante se encuentre matriculado en un programa formal de estudios, aunque también puede dedicarse a la búsqueda de conocimientos de manera autónoma o informal.

Existen diversas clasificaciones o tipos de estudiante, que se establecen a partir del modelo de enseñanza, la dedicación temporal que implica el estudio, el plan académico en el que se inscribe y otras características. La palabra estudiante suele ser utilizada como sinónimo de alumno. Este concepto hace referencia a aquellos individuos que aprenden de otras personas. El término alumno proviene del latín *alumnum*, que a su vez deriva de *alere* ("alimentar").

### **2.2.2.2 DOCENTE UNIVERSITARIO**

Un docente es aquel individuo que se dedica a enseñar o que realiza acciones referentes a la enseñanza. La palabra deriva del término latino *docens*, que a su vez procede de *docēre* (traducido al español como "enseñar").

El docente, en definitiva, reconoce que la enseñanza es su dedicación y profesión fundamental. Por lo tanto, sus habilidades consisten en enseñar de la mejor forma posible a quien asume el rol de educando, más allá de la edad o condición que éste posea.

Además de establecer que existen docentes en diversos niveles también hay que subrayar que estos también pueden clasificarse en función de las asignaturas o materias que impartan. De esta manera, podemos encontrar docentes especializados en Matemáticas, en Lengua y Literatura, en Geografía e Historia, en Biología o en Física y Química, entre otros.

La docencia, entendida como enseñanza, es una actividad realizada a través de la interacción de tres elementos: el docente, sus alumnos y el objeto de conocimiento. Una concepción teórica e idealista supone que el docente tiene la obligación de transmitir sus saberes al alumno mediante diversos recursos, elementos, técnicas y herramientas de apoyo. Así, el docente asume el rol de

fuerza de conocimientos y el educando se convierte en un receptor ilimitado de todo ese saber. En los últimos tiempos, este proceso es considerado como más dinámico y recíproco.

### **2.2.2.3 POSTULANTE**

Es la persona que, habiendo completado sus estudios secundarios, o que estando por completarlos en el año previo al inicio del año académico en la universidad, desea seguir sus estudios superiores en una Universidad.

### **2.2.3 HUELLA DIGITAL**

“La comparación de la huella digital es una de las técnicas más antiguas y ampliamente utilizadas y aceptas a nivel global. La huella digital aparece generalmente constituida por una serie de líneas oscuras que representan las crestas y una serie de espacios blancos que representan los valles. La identificación con huellas digitales está basada principalmente en las minucias (la ubicación y dirección de las terminaciones de crestas, bifurcaciones, deltas, valles y crestas, aunque existen muchas otras características de huellas digitales.” (Galvis, 2007).

Como afirma Camgal (s.f.), un huella digital son aquellas ondas que se encuentran situadas en las yemas de los dedos y al ponerlas sobre una superficie plana se quedan marcadas, para esto casi siempre se pone el dedo índice o el pulgar, esto sin duda es una característica individual de cada persona que sin duda sirve para identificar a las personas, todos tenemos crestas diferentes es por eso que es individual e inconfundible.

“Las huellas digitales son los relieves epidérmicos que se encuentran en las falangetas de cada uno de nuestros dedos que son también una característica física única que distingue a todos los seres humanos y son irrepetibles.” (Rendón, s.f.).

La huella digital, se describe básicamente como una sucesión de crestas

separadas entre sí por valles. La disposición de estas crestas se traduce en la aparición de ciertos puntos singulares denominados terminaciones y bifurcaciones. Este grupo de puntos singulares recibió el nombre de minucias y, mediante su orientación, localización y tipo se puede identificar a una persona. Estudios realizados a finales del siglo XIX revelaron que la estructura de la huella digital no pertenece a las capas más superficiales de la piel, si no que se trata de algo intrínseco. Por ello, si por cualquier circunstancia se pierde la piel de un dedo, el recrecimiento de la piel provocará que la huella se vuelva a reconstruir tal y como era antes. Además se ha comprobado que cada huella posee un elevado grado de unicidad, llegando al punto de que un mismo individuo tiene huellas totalmente diferentes en cada uno de sus dedos. Sin embargo, no solo las minucias sirven para realizar la identificación de un individuo. Existen otros puntos singulares como los llamados núcleo (core), y delta, que son muy utilizados para implementar sistemas de reconocimiento. El núcleo podría describirse como el punto en el que la orientación de las crestas tiende a converger, mientras que los deltas serían los puntos donde el flujo de crestas presenta una divergencia.

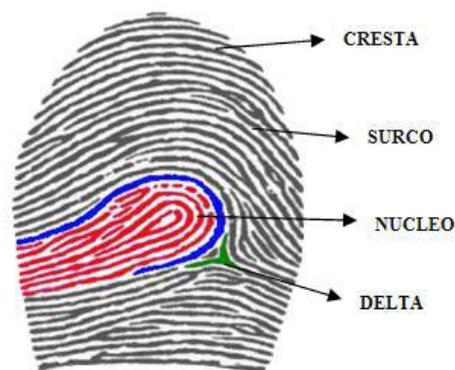


Figura N° 2.1: Puntos singulares de una huella dactilar (Vargas, 2013)

## **PROCESO DE COMPARACIÓN DE HUELLAS DIGITALES**

Debido a que el proceso de comparación de patrones de huellas, en el bloque del sistema denominado Matcher, supone que nunca va a haber una coincidencia exacta entre patrones de huella como es natural, los algoritmos de comparación de las huellas implementados en el Matcher son de carácter

estadístico para lo que se define un parámetro de puntuación de la coincidencia, o Matching Score (parámetro probabilístico en el rango [0..1]); entonces, si éste supera cierto umbral definido el sistema tomará la decisión de aceptar la coincidencia entre patrones de huella resultando en una autenticación positiva del individuo. Precisamente, por tratarse de un proceso de comparación estocástico, como no puede ser de otra manera, nunca habrá certeza en la coincidencia de huellas, es decir, Matching Score igual a 1. Esto significa, entonces, que el sistema cometerá errores de autenticación.

Como afirma Arenas (s.f.), el proceso de comparación uno a uno, el usuario presenta su(s) dato(s) biométrico(s) y este se compara con la plantilla biométrica almacenada en una base de datos o en un dispositivo portátil, verificando si hay o no coincidencia para esa identidad en la referencia establecida. Una forma gráfica de representar el sistema operativo de las pruebas biométricas de la actualidad es el que a continuación se muestra:

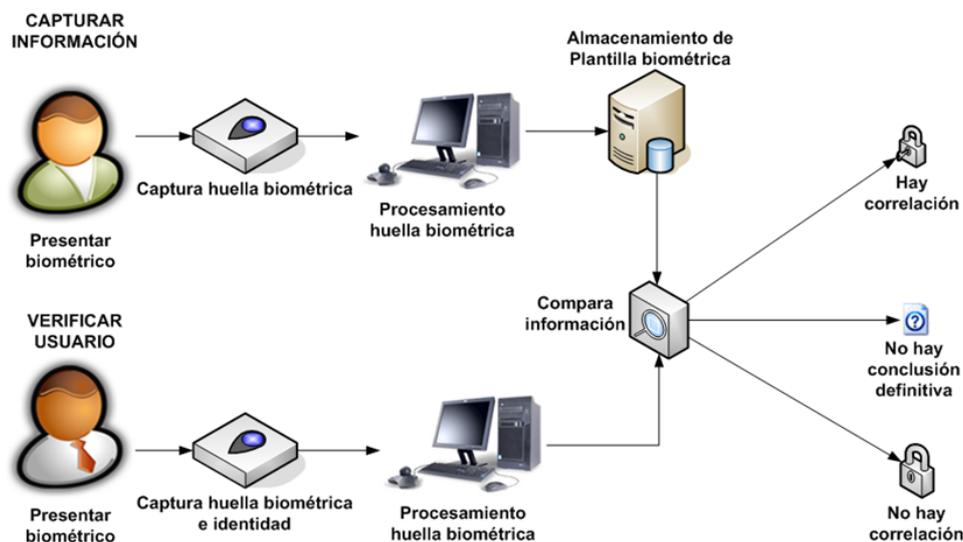


Figura N° 2.2: Proceso de comparación de huellas digitales (Arenas, s.f.)

### 2.2.3.1 RENDIMIENTO BIOMÉTRICO

“El rendimiento de una medida biométrica se define generalmente en términos de tasa de falso positivo (False Acceptance Rate o FAR), la tasa de falso negativo (False NonMatch Rate o FNMR, también False Rejection Rate o

FRR), y el fallo de tasa de alistamiento (Failure-to-enroll Rate, FTR o FER). En los sistemas biométricos reales el FAR y el FRR pueden transformarse en los demás cambiando cierto parámetro. Una de las medidas más comunes de los sistemas biométricos reales es la tasa en la que el ajuste en el cual acepta y rechaza los errores es igual: la tasa de error igual (Equal Error Rate o EER), también conocida como la tasa de error de cruce (Cross-over Error Rate o CER). Cuanto más bajo es el EER o el CER, se considera que el sistema es más exacto." (Wikipedia, 2011).

"Rendimiento del Sistema ("throughput rate"), velocidad a la que el sistema puede identificar o autenticar a los individuos. Hoy en día se consideran aceptables en valores de unos 10 individuos por minuto." (Gómez, 2007).

Según Bolívar y Cortez (2009), en cuanto al rendimiento de los sistemas biométricos, estos están definidos por parámetros biométricos, que miden la efectividad de un sistema biométrico de identificación y verificación, a continuación se explican los parámetros.

### **FAR (False Acceptance Rate)**

Hace referencia a la tasa de falso positivo, en otras palabras es la probabilidad de que un usuario no autorizado sea aceptado. Este parámetro se ajusta para evitar los fraudes en los sistemas biométricos.

### **FRR (False Rejection Rate)**

Es la tasa de rechazo erróneo, es decir la probabilidad de que un usuario que está autorizado sea rechazado a la hora de intentar acceder al sistema. Si esto ocurre frecuentemente se podrá llegar a concluir que el sistema no está funcionando correctamente y por tanto deberá ser revisado.

### **FER (Failure to Enroll Rate)**

Hace referencia a la tasa de fallo de alistamiento, es decir los usuarios que son rechazados cuando van a ser registrados es por causa de la mala calidad de su muestra.

## UMBRAL

Es un tipo de referencia, es la puntuación que determina la consistencia de un patrón. Éste se ajusta dependiendo del nivel de seguridad que se use.

## FTE (Failure to Enroll)

Este se encarga de indicar la probabilidad numérica de que alguien no sea registrado a causa de un fallo.

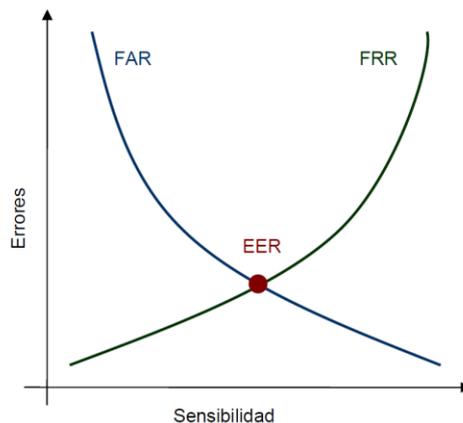


Figura N° 2.3: Tasas de error de un sistema biométrico (Lindoso, 2009).

### 2.2.4 LECTOR DE HUELLA DIGITAL

Es un dispositivo de seguridad encargado de detectar los relieves del dedo por medio de luz o por medio de sensores eléctricos, posteriormente genera una imagen digital la cuál es enviada a la computadora y almacenada en una base de datos en los que se le asocia con la información de una persona. Cada vez que se coloca el dedo sobre la superficie óptica del lector, este envía la información y la computadora determina a que persona corresponde o si se trata de alguien no identificado. El nombre que se le da en Inglés es ("Finger Print Reader"), lo que traducido al español significa lector de impresión de dedo, otro modo de llamarlo es control biométrico.

### Características generales del lector de huella digital

- a) Si se trata de lectores de huella digital independientes, tiene la capacidad de almacenar información sobre las personas, mientras que uno no independiente, envía la información a la computadora y esta se encarga de

guardar la información.

- b) Tienen un tiempo exploración, el cuál determina cuánto tarda en realizar la lectura de la huella digital, se mide en segundos y puede ser de hasta 1.2 s.
- c) Tienen un tiempo de verificación, el cuál determina cuánto tarda en procesar la información que recabe de la huella digital, este se encuentra en promedio, se mide en segundos y puede ser de hasta 1.5 s.
- d) Algunos equipos independientes incluso pueden tener la opción de insertar una contraseña como medida de seguridad adicional.
- e) Tienen dos valores llamados porcentajes de aceptación y rechazo falsos, las cuáles determinan la fiabilidad del dispositivo, este se mide en % y puede ser muy bajo como ejemplo 0.001%.

Según López (s.f.) las huellas se obtienen mediante la adquisición directa de la huella digital al colocar el dedo sobre la superficie sensible del sensor electrónico. El procedimiento de la conversión de la huella capturada en una imagen digital depende de los principios físicos de funcionamiento del sensor utilizado. Atendiendo a estos principios físicos, puede establecerse la siguiente clasificación de sensores:

- a) Sensores ópticos.** Entre estos sensores están aquellos que se basan en la reflexión de la luz sobre la yema del dedo (FTIR, Frustrated Total Internal Reflexion), los sensores basados en fibra óptica, los electro-ópticos y los sensores sin contacto.
- b) Sensores de estado sólido.** A este grupo pertenecen los sensores capacitivos, térmicos, de campo eléctrico y piezoeléctrico.
- c) Sensores ultrasónicos.** Estos dispositivos funcionan proyectando pulsos ultrasónicos.

### **Sensores basados en FTIR**

La técnica de captura FTIR es la más antigua y también la más utilizada. En el momento en el que el dedo se apoya sobre la superficie de cristal del sensor (prisma), un diodo LED proyecta un haz de luz difusa por debajo del cristal. La luz que atraviesa el prisma e incide sobre las crestas de la huella se dispersa,

reflejándose de manera aleatoria en múltiples direcciones. La luz que incide en el interior de la estructura de crestas (valles) se refleja en una determinada dirección (reflexión total). Esta luz direccional es focalizada mediante un sistema de lentes hacia un dispositivo CCD o CMOS, capturándose así la imagen de la huella digital. Recientemente, se ha desarrollado una variante de esta técnica en la que se sustituye el prisma de cristal por una lámina de pequeños prismas distribuidos a lo largo de la superficie sensible. La calidad de las imágenes adquiridas es ligeramente menor, pero tiene la ventaja de que el tamaño del dispositivo se reduce significativamente.



Figura N° 2.4: Lector de huella óptico digitalPersona U.are.U 4000B (crossmatch, s.f.)

### **2.2.5 METODOLOGÍA DE DESARROLLO ÁGIL DE PROGRAMACIÓN EXTREMA**

“La programación extrema (XP) es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas. Este modelo de programación se basa en una serie de metodologías de desarrollo de software en la que se da prioridad a los trabajos que dan un resultado directo y que reducen la burocracia que hay alrededor de la programación. Una de las características principales de este método de programación, es que sus ingredientes son conocidos desde el principio de la informática. Los autores de XP han seleccionado aquellos que han considerado mejores y han profundizado en sus relaciones y en cómo se refuerzan los unos con los otros. El resultado de esta selección ha sido esta metodología única y compacta. Por esto, aunque no está basada en principios nuevos, sí que el

resultado es una nueva manera de ver el desarrollo de software. El objetivo que se perseguía en el momento de crear esta metodología era la búsqueda de un método que hiciera que los desarrollos fueran más sencillos. Aplicando el sentido común.” (Aguilar, 2002).

Como afirma Berrueta (2006), la programación extrema es una metodología para el desarrollo de proyectos informáticos que trata de dar solución a los problemas de la ingeniería del software desde un enfoque completamente distinto al que ha venido siendo habitual. Los estudios demuestran que la mayoría de proyectos de software fracasan, porque exceden sus plazos, superan su presupuesto, no se ajustan a las auténticas necesidades del cliente, presentan una calidad deficiente o, en muchos casos, son abortados. Las metodologías tradicionales han tratado de poner coto a esta situación mediante un control intensivo del proceso. Al hacerlo, se está ignorando que las necesidades del cliente y sus expectativas realmente cambian durante el desarrollo del proyecto. Como respuesta, ha surgido una nueva familia de metodologías denominadas ágiles, cuyo rasgo principal consiste en contemplar y dar respuesta a las necesidades dinámicas del cliente. De entre las metodologías ágiles, la que goza de mayor popularidad es la programación extrema, propuesta en 1999 por Kent Beck, en un libro titulado precisamente “abrazo el cambio”. La programación extrema recibe este calificativo precisamente porque defiende un enfoque radical. Reconoce las bondades de las prácticas de las metodologías tradicionales (diseño, pruebas, revisiones de código, etc.) y propone llevarlas hasta su extremo: “si diseñar es bueno, diseñemos todo el tiempo”, “si las pruebas son buenas, probemos todo el tiempo”, etc.

Según Carrillo y Pérez (2008), los roles de la Programación Extrema (XP), son los siguientes:

- a) Programador:** El programador escribe las pruebas unitarias y produce el código del sistema.
- b) Cliente:** Escribe las historias de los usuarios y las pruebas funcionales

para validar su implementación. El cliente da una gran prioridad a las historias de usuarios y decide cual implementar en cada iteración centrándose en aportar mayor valor al negocio.

- c) Encargado de Pruebas (Tester):** Ayuda al cliente a escribir las pruebas funcionales. Se encarga de ejecutar las pruebas con regularidad, difunde los resultados obtenidos al equipo y es el responsable de las herramientas que dan soporte a las pruebas.
- d) Encargado de Seguimiento (Tracker):** Es el que proporciona la realimentación al equipo. Realiza el seguimiento del proceso de cada iteración y verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado en ello para la mejora de futuras estimaciones.
- e) Entrenador (Coach):** Es el responsable del proceso global. Se encarga de proveer guías al equipo de forma que se apliquen las practicas XP y se vaya siguiendo el proceso correctamente.
- f) Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema que es necesario para el proyecto, en el que surjan problemas.
- g) Gestor (Big boss):** Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es la de coordinación.

Erljman y Goyén (s.f.), describen las 12 prácticas fundamentales de la metodología XP, las cuales se detallan a continuación:

- a) El Juego de la Planificación,** es la práctica que define la forma general de trabajar. Está compuesta por la Planificación del "Release" y por la Planificación de la Iteración. En la Planificación del "Release" se define qué es lo que se pretende tener como producto en un período de 4 o 6 meses. El cliente<sup>4</sup> define una gran cantidad de requerimientos, llamadas Historias, que son los que le dan mayor valor al negocio y que deben ser implementados dentro de ese período. Una vez definido el conjunto de Historias, éstas son analizadas y estimadas por el grupo de programadores para que finalmente el cliente las ordene en función de

su valor. Cuando se tienen ordenadas las Historias, se procede a elegir aquellas cuya suma del tiempo de desarrollo no supere el período del "release".

**b) En la Planificación de la Iteración,** se definen las actividades para las siguientes 3 o 4 semanas. Teniendo en cuenta la capacidad productiva del grupo de desarrollo para la iteración, denominada Velocidad, el cliente elige el conjunto de Historias de mayor valor para que sean implementadas en la iteración planeada. A continuación, los programadores dividen las Historias en tareas más pequeñas, denominadas Tareas de Ingeniería. Luego, cada programador elige las tareas que desea implementar, las analiza en mayor detalle y realiza una estimación de su tiempo de desarrollo. Finalmente, el cliente ordena en función de sus necesidades las Historias estimadas, dejando para iteraciones posteriores aquellas que sobrepasen la capacidad productiva de la iteración. El Juego de la Planificación del "Release" y de la Iteración se puede resumir en la siguiente figura:

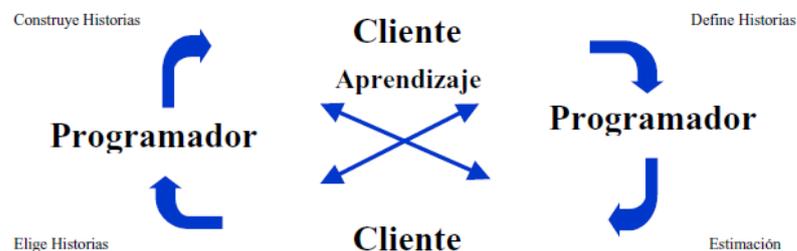


Figura N° 2.5: Juego de Planificación (Erljman y Goyén, s.f.)

**c) Además de la mecánica del proceso explicada anteriormente,** es importante hacer notar el aprendizaje que tienen los programadores en la realización de estimaciones al tener que repetirlas en períodos cortos de tiempo. También es importante resaltar el aprendizaje que tiene el cliente en la definición de las Historias, para que sean más claras para los programadores.

**d) Las Entregas Frecuentes,** permiten que el sistema empiece con algo simple y se ponga en producción rápidamente, para luego evolucionar a

través de actualizaciones e incorporación de funcionalidad frecuente. Estas actualizaciones son realizadas en base a las prioridades establecidas por el cliente durante la Planificación de la Iteración. Las entregas frecuentes se dividen en períodos denominados iteraciones. Se recomienda que las iteraciones sean cortas y que no duren más de 3 o 4 semanas.

- e) La Metáfora,** es una descripción general del sistema, que se establece al comenzar el proyecto, que fortifica la integridad conceptual, ayuda a guiar el proceso de desarrollo y mantiene una visión unificada entre los actores. La Metáfora determina un estándar en el vocabulario que será utilizado por los programadores y el cliente, que luego ayudará a establecer las clases y métodos del sistema.
- f) Los Diseños Simples,** hacen que los sistemas desarrollados con XP sean creados de la manera más sencilla, pero cumpliendo con la funcionalidad que el cliente especificó en el Juego de la Planificación. XP le resta importancia a las necesidades desconocidas y especulativas del futuro y sólo atiende las necesidades actuales del cliente. Cabe aclarar que esto no quiere decir que los diseños sean de baja calidad, sino que se empieza por lo más sencillo que funcione y luego se transforma en algo más complejo si el diseño demuestra insuficiencias. La complejidad innecesaria debe ser eliminada ni bien se descubra.
- g) El Testing Continuo,** exige que los equipos XP validen el funcionamiento del software en todo momento. XP define dos tipos de test. Por un lado, los programadores diseñan y ejecutan los Test de Unidad previo a la implementación, mientras que el cliente diseña y ejecuta los Test de Aceptación. Los Test de Aceptación le permiten al cliente asegurarse que se ha desarrollado la funcionalidad negociada durante el Juego de la Planificación. Cada funcionalidad del sistema (Historia) debe tener por lo menos un Test de Aceptación asociado.
- h) El Refactoring,** se define como "el proceso de alterar un sistema computacional de tal forma de mejorar su estructura interna sin alterar el comportamiento externo". La incorporación de esta práctica, permite

que los diseños del sistema se vayan perfeccionando continuamente durante todo el proceso de desarrollo, sin atarse a un diseño preliminar rígido como en el caso de las metodologías tradicionales. A diferencia de otras metodologías, XP acepta que en realidad lo único constante es el cambio y se adapta a coexistir junto a él. Aplicando esta práctica de forma continua, XP apunta a que el software se pueda mejorar y modificar con facilidad.

- i) La Programación en Pareja**, exige que toda la programación y los test se realicen de a dos programadores por computadora. Hay experimentos, que demuestran que la programación en pareja produce mejor software a un costo igual o menor que la programación individual.
- j) La Propiedad Colectiva del Código**, hace que ninguna porción del código tenga programadores "dueños". Esto aumenta la velocidad de desarrollo ya que cuando se necesita algún cambio, cualquier programador lo puede hacer sin depender de los otros.
- k) La Integración Continua**, indica que los equipos XP deben integrar el software construido diariamente. Esto minimiza el riesgo de enfrentar severos problemas de integración, vistos en proyectos que no integran con frecuencia.
- l) Para mantener al equipo saludable**, descansado y aumentar la productividad y la efectividad, XP propone Semanas de 40 Horas de Trabajo.
- m) La Presencia del cliente On-Site permite que el proyecto sea guiado por un individuo dedicado**, con el poder de decisión necesario para determinar los requerimientos y las prioridades de entrega. El efecto de la presencia continua en el lugar de desarrollo, hace que la comunicación sea fluida, con menos necesidades de documentación por escrito y permite resolver rápidamente las dudas y decisiones que puedan aparecer.
- n) Para que un equipo pueda trabajar de forma efectiva y pueda compartir el código de forma colectiva**, los programadores deben ponerse de acuerdo en establecer un estilo en común mediante una serie

de reglas que permitan Estandarizar el Estilo de Programación. Esto incluye la estandarización de nomenclaturas de variables, formato común para comentarios dentro del código, etc.

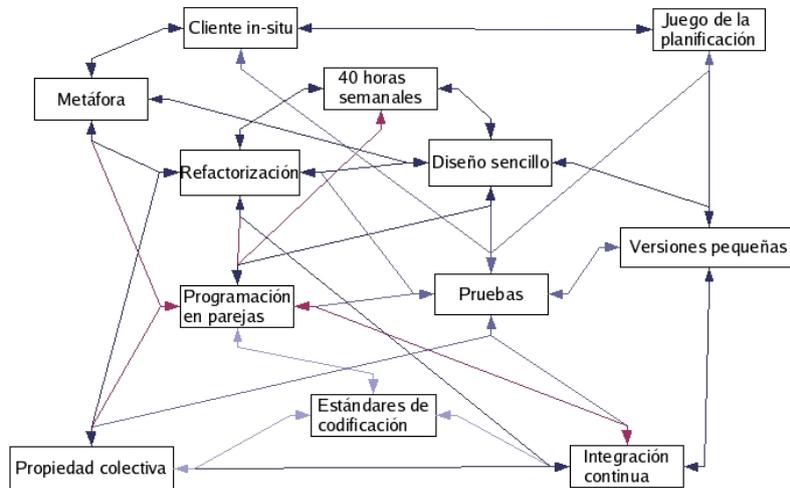


Figura N° 2.6: Refuerzo de las Prácticas de XP (Letelier y Penadés, s.f.)

Las actividades de XP, se describen a continuación.

- a) **Codificar**, es necesario codificar y plasmar nuestras ideas a través del código. En programación, el código expresa la interpretación del problema, así podemos utilizar el código para comunicar, para hacer comunes las ideas, y por tanto para aprender y mejorar.
- b) **Hacer pruebas**, las características del software que no pueden ser demostradas mediante pruebas simplemente no existen. Las pruebas dan la oportunidad de saber si lo implementado es lo que en realidad se tenía en mente. Las pruebas nos indican que nuestro trabajo funciona, cuando no podemos pensar en ninguna prueba que pudiese originar un fallo en nuestro sistema, entonces habremos acabado por completo.
- c) **Escuchar**, “los programadores no lo conocemos todo, y sobre todo muchas cosas que las personas de negocios piensan que son interesantes. Si ellos pudieran programarse su propio software ¿para qué nos querrían?”. Si vamos a hacer pruebas tenemos que preguntar si lo obtenido es lo deseado, y tenemos que preguntar a quién necesita la información. Tenemos que escuchar a nuestros clientes cuáles son los problemas de su negocio, debemos de tener una escucha activa explicando lo que es fácil y

difícil de obtener, y la realimentación entre ambos nos ayudan a todos a entender los problemas.

**d) Diseñar,** el diseño crea una estructura que organiza la lógica del sistema, un buen diseño permite que el sistema crezca con cambios en un solo lugar. Los diseños deben de ser sencillos, si alguna parte del sistema es de desarrollo complejo, lo apropiado es dividirla en varias. Si hay fallos en el diseño o malos diseños, estos deben de ser corregidos cuanto antes.

Resumiendo las actividades de XP: Tenemos que codificar porque sin código no hay programas, tenemos que hacer pruebas porque sin pruebas no sabemos si hemos acabado de codificar, tenemos que escuchar, porque si no escuchamos no sabemos que codificar ni probar, y tenemos que diseñar para poder codificar, probar y escuchar indefinidamente.

## ARTEFACTOS XP

A continuación describimos los artefactos de XP, entre los que se encuentran: Historias de Usuario, Tareas de Ingeniería y Tarjetas CRC.

### Historias de Usuario

Representan una breve descripción del comportamiento del sistema, emplea terminología del cliente sin lenguaje técnico, se realiza una por cada característica principal del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación.

<b>Historia de Usuario</b>	
<b>Número:</b>	<b>Nombre Historia de Usuario:</b>
<b>Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):</b>	
<b>Usuario:</b>	<b>Iteración Asignada:</b>
<b>Prioridad en Negocio:</b> (Alta / Media / Baja)	<b>Puntos Estimados:</b>
<b>Riesgo en Desarrollo:</b> (Alto / Medio / Bajo)	<b>Puntos Reales:</b>
<b>Descripción:</b>	
<b>Observaciones:</b>	

Tabla N° 2.1. Modelo propuesto para una historia de usuario (Anaya, s.f.)

Estas deben proporcionar sólo el detalle suficiente como para poder hacer razonable la estimación de cuánto tiempo requiere la implementación de la historia, difiere de los casos de uso porque son escritos por el cliente, no por los programadores, empleando terminología del cliente. "Las historias de usuario son más "amigables" que los casos de uso formales".

Las Historias de Usuario tienen tres aspectos:

- a) Tarjeta: en ella se almacena suficiente información para identificar y detallar la historia.
- b) Conversación: cliente y programadores discuten la historia para ampliar los detalles (verbalmente cuando sea posible, pero documentada cuando se requiera confirmación)
- c) Pruebas de Aceptación: permite confirmar que la historia ha sido implementada correctamente.

<b>Caso de Prueba de Aceptación</b>	
<b>Código:</b>	<b>Historia de Usuario (Nro. y Nombre):</b>
<b>Nombre:</b>	
<b>Descripción:</b>	
<b>Condiciones de Ejecución:</b>	
<b>Entrada / Pasos de ejecución:</b>	
<b>Resultado Esperado:</b>	
<b>Evaluación de la Prueba:</b>	

Tabla N° 2.2. Modelo propuesto para una prueba de aceptación (Anaya, s.f.)

## Task Card

<b>Tarea de Ingeniería</b>	
<b>Número Tarea:</b>	<b>Historia de Usuario (Nro. y Nombre):</b>
<b>Nombre Tarea:</b>	
<b>Tipo de Tarea :</b> Desarrollo / Corrección / Mejora / Otra (especificar)	<b>Puntos Estimados:</b>
<b>Fecha Inicio:</b>	<b>Fecha Fin:</b>
<b>Programador Responsable:</b>	
<b>Descripción:</b>	

Tabla N° 2.3. Modelo propuesto para una tarea de ingeniería (Anaya, s.f.)

### **Tarjetas CRC** (Clase - Responsabilidad – Colaborador).

Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. En la siguiente figura se muestra cómo se distribuye esta información.

Nombre de la clase.	
Responsabilidades	Colaboradores

Tabla N°.2.4. Modelo de tarjeta CRC (Anaya, s.f.)

Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

En la práctica conviene tener pequeñas tarjetas de cartón, que se llenarán y que son mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas.

Los pasos a seguir para llenar las tarjetas son los siguientes:

- a) Encontrar clases
- b) Encontrar responsabilidades
- c) Definir colaboradores
- d) Disponer las tarjetas

Para encontrar las clases debemos pensar qué cosas interactúan con el sistema (en nuestro caso el usuario), y qué cosas son parte del sistema, así como las pantallas útiles a la aplicación (un despliegue de datos, una entrada de parámetros y una pantalla general, entre otros). Una vez que las clases principales han sido encontradas se procede a buscar los atributos y las

responsabilidades, para esto se puede formular la pregunta ¿Qué sabe la clase? y ¿Qué hace la clase? Finalmente se buscan los colaboradores dentro de la lista de clases que se tenga.

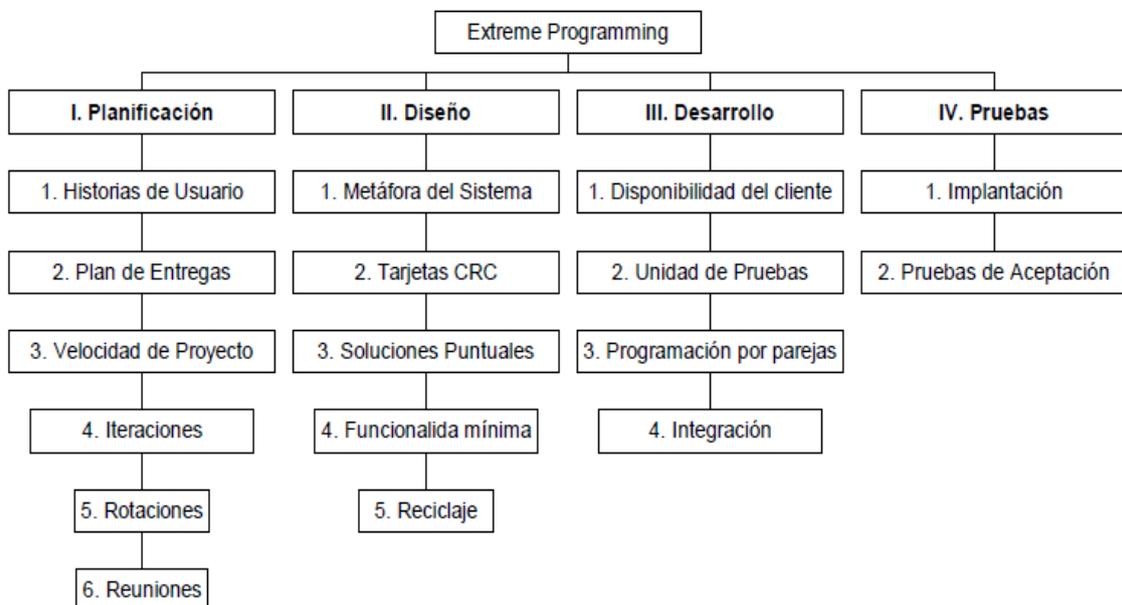


Figura N° 2.7: Fases de la programación extrema (Fernández, 2002)

## 2.2.6 SISTEMA GESTOR DE BASE DE DATOS RELACIONAL

Como afirma Marin (2006), se define un Sistema Gestor de Bases de Datos o SGBD, también llamado DBMS (Data Base Management System) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos o BD, (DB Data Base).

Antes de aparecer los SGBD (década de los setenta), la información se trataba y se gestionaba utilizando los típicos sistemas de gestión de archivos que iban soportados sobre un sistema operativo. Éstos consistían en un conjunto de programas que definían y trabajaban sus propios datos. Los datos se almacenan en archivos y los programas manejan esos archivos para obtener la información. Si la estructura de los datos de los archivos cambia, todos los programas que los manejan se deben modificar; por ejemplo, un programa trabaja con un archivo de datos de alumnos, con una estructura o registro ya

definido; si se incorporan elementos o campos a la estructura del archivo, los programas que utilizan ese archivo se tienen que modificar para tratar esos nuevos elementos. En estos sistemas de gestión de archivos, la definición de los datos se encuentra codificada dentro de los programas de aplicación en lugar de almacenarse de forma independiente, y además el control del acceso y la manipulación de los datos vienen impuesto por los programas de aplicación. Esto supone un gran inconveniente a la hora de tratar grandes volúmenes de información. Surge así la idea de separar los datos contenidos en los archivos de los programas que los manipulan, es decir, que se pueda modificar la estructura de los datos de los archivos sin que por ello se tengan que modificar los programas con los que trabajan. Se trata de estructurar y organizar los datos de forma que se pueda acceder a ellos con independencia de los programas que los gestionan.

Como afirma Marqués (s.f), los sistemas de gestión de la base de datos (en adelante SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Se denomina sistema de bases de datos al conjunto formado por la base de datos, el SGBD y los programas de aplicación que dan servicio a la empresa u organización.

El modelo seguido con los sistemas de bases de datos es muy similar al modelo que se sigue en la actualidad para el desarrollo de programas con lenguajes orientados a objetos, en donde se da una implementación interna de un objeto y una especificación externa separada. Los usuarios del objeto sólo ven la especificación externa y no se deben preocupar de cómo se implementa internamente el objeto. Una ventaja de este modelo, conocido como abstracción de datos, es que se puede cambiar la implementación interna de un objeto sin afectar a sus usuarios ya que la especificación externa no se ve alterada. Del mismo modo, los sistemas de bases de datos separan la definición de la estructura física de los datos de su estructura lógica, y almacenan esta definición en la base de datos. Todo esto es gracias a la existencia del SGBD,

que se sitúa entre la base de datos y los programas de aplicación.

Generalmente, un SGBD proporciona los servicios que se citan a continuación:

- a. El SGBD permite la definición de la base de datos mediante un lenguaje de definición de datos. Este lenguaje permite especificar la estructura y el tipo de los datos, así como las restricciones sobre los datos.
- b. El SGBD permite la inserción, actualización, eliminación y consulta de datos mediante un lenguaje de manejo de datos. El hecho de disponer de un lenguaje para realizar consultas reduce el problema de los sistemas de ficheros, en los que el usuario tiene que trabajar con un conjunto fijo de consultas, o bien, dispone de un gran número de programas de aplicación costosos de gestionar. Hay dos tipos de lenguajes de manejo de datos: los procedurales y los no procedurales. Estos dos tipos se distinguen por el modo en que acceden a los datos. Los lenguajes procedurales manipulan la base de datos registro a registro, mientras que los no procedurales operan sobre conjuntos de registros. En los lenguajes procedurales se especifica qué operaciones se debe realizar para obtener los datos resultado, mientras que en los lenguajes no procedurales se especifica qué datos deben obtenerse sin decir cómo hacerlo. El lenguaje no procedural más utilizado es el SQL (Structured Query Language) que, de hecho, es un estándar y es el lenguaje de los SGBD relacionales.
- c. El SGBD proporciona un acceso controlado a la base de datos mediante:
  - c.1. Un sistema de seguridad, de modo que los usuarios no autorizados no puedan acceder a la base de datos.
  - c.2. Un sistema de integridad que mantiene la integridad y la consistencia de los datos.
  - c.3. Un sistema de control de concurrencia que permite el acceso compartido a la base de datos.
  - c.4. Un sistema de control de recuperación que restablece la base de datos después de que se produzca un fallo del hardware o del software.
  - c.5. Un diccionario de datos o catálogo, accesible por el usuario, que

contiene la descripción de los datos de la base de datos.

A diferencia de los sistemas de ficheros, en los que los programas de aplicación trabajan directamente sobre los ficheros de datos, el SGBD se ocupa de la estructura física de los datos y de su almacenamiento. Con esta funcionalidad, el SGBD se convierte en una herramienta de gran utilidad. Sin embargo, desde el punto de vista del usuario, se podría discutir que los SGBD han hecho las cosas más complicadas, ya que ahora los usuarios ven más datos de los que realmente quieren o necesitan, puesto que ven la base de datos completa. Conscientes de este problema, los SGBD proporcionan un mecanismo de vistas que permite que cada usuario tenga su propia vista o visión de la base de datos. El lenguaje de definición de datos permite definir vistas como subconjuntos de la base de datos.

Todos los SGBD no presentan la misma funcionalidad, depende de cada producto. En general, los grandes SGBD multiusuario ofrecen todas las funciones que se acaban de citar e incluso más. Los sistemas modernos son conjuntos de programas extremadamente complejos y sofisticados, con millones de líneas de código y con una documentación consistente en varios volúmenes. Lo que se pretende es proporcionar un sistema que permita gestionar cualquier tipo de requisitos y que tenga un 100 % de fiabilidad ante cualquier tipo de fallo. Los SGBD están en continua evolución, tratando de satisfacer los requisitos de todo tipo de usuarios. Por ejemplo, muchas aplicaciones de hoy en día necesitan almacenar imágenes, vídeo, sonido, etc. Para satisfacer a este mercado, los SGBD deben evolucionar. Conforme vaya pasando el tiempo, irán surgiendo nuevos requisitos, por lo que los SGBD nunca permanecerán estáticos.

Hansen y Hansen (2004) señalan que: Un sistema de gestión de base de datos o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir, manipular y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto

debe permitir:

- a. Definir una base de datos: consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.
- b. Construir una base de datos: es el proceso de almacenar los datos sobre algún medio de almacenamiento.
- c. Manipular una base de datos: incluye funciones como consulta, actualización, etc. de bases de datos.

Algunas de las características deseables en un Sistema Gestor de base de datos SGBD son:

- a. Control de la redundancia: La redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco, puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.
- b. Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- c. Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

## **COMPONENTES SGBD**

- a. Lenguajes
  - a.1. Lenguaje de definición de datos (DDL)
  - a.2. Lenguaje de manipulación de datos (DML)
- b. Diccionario de datos: lugar donde se deposita información sobre todos los objetos que forman la base de datos (estructura lógica y física de los datos, definiciones de todos los objetos de la base de datos...)
- c. Seguridad e integridad de datos
  - c.1. Debe garantizarse la protección de los datos contra accesos no autorizados
  - c.2. Los SGBD deben ofrecer mecanismos para implantar restricciones de integridad

- c.3. Proporciona herramientas y mecanismos para planificación y realización de copias de seguridad
- c.4. Debe ser capaz de recuperar la BD llevándola a un estado consistente
- c.5. Debe asegurar el acceso concurrente
- d. Usuarios de los SGBD
  - d.1. Usuarios de la categoría DBA (administradores)
  - d.2. Usuarios de tipo RESOURCE, que pueden crear sus propios objetos y tener acceso a los objetos sobre los que se les ha concedido permisos.
  - d.3. Usuarios de tipo CONNECT, que solamente pueden utilizar los objetos a los que se les ha concedido permiso
- e. Todos los SGBD proporcionan una serie de herramientas de administración.
- f. Permite a los administradores la gestión de la BD y gestión de usuarios y permisos, entre otros.

### **2.2.7 LENGUAJE DE PROGRAMACIÓN ORIENTADO A OBJETOS**

Como afirma Effi (2004), los lenguajes de programación orientada a objetos han existido por muchos años, pero sólo recientemente han despertado interés. Los lenguajes de OOP más populares son SmallTalk, C++ y Java. Otro lenguaje popular, como Visual Basic, permite al programador usar objetos gráficos pero no llena todos los requisitos de un verdadero lenguaje OOP. Por ejemplo al mover un icono a otra aplicación, se desplaza el código asociado a él. Algunos lenguajes de OOP están diseñados específicamente para usarse en el desarrollo de interfaces gráficas de usuario (GUI, Graphical User Interfaces). Entre los elementos que las integran se incluyen ventanas, iconos, cuadros de desplazamiento y otras imágenes gráficas que ayudan al usuario a interactuar con el programa con un mínimo de esfuerzo.

“La programación orientada a objetos es realmente un nuevo estilo de programación, el cual, básicamente consiste en definir clases y poner dichas

clases a comunicarse o conversar entre si.” (Flores, 2005).

“La programación a orientada a objetos está basado a una filosofía distinta a las anteriores estilos de programación, la idea fundamental de la programación se mantiene, es necesario que a la hora de resolver problemas utilizando orientación a objetos se piense de manera diferente.” (Osorio, 2004).

Según Grady Booch, se define a la programación orientada a objetos (POO) como: un método de implementación en la que los programas se organizan como colecciones de objetos, cada uno de los cuales representan una instancia de alguna clase.

A continuación se muestran conceptos de: objeto, métodos, mensajes y clases, relacionadas a la programación orientada a objetos:

- a) OBJETO:** Objeto es una colección de elementos de datos junto con funciones asociadas utilizada para operar sobre esos datos. Los objetos de un programa se comunican con cada uno de los restantes pasando mensajes.
- b) MÉTODOS:** Los procedimientos y funciones, denominados Métodos, residen en el objeto y determinan cómo actúan los objetos cuando reciben un mensaje, un Método es el procedimiento que se invoca para actuar sobre un objeto.
- c) MENSAJE:** Un mensaje es la acción que hace un Objeto.
- d) CLASES:** Una Clase es la descripción de un conjunto de objetos, este consta de métodos y datos que resumen características de un conjunto de objetos, por lo que podemos decir que una clase es la declaración de un tipo objeto. Cada vez que se construye un objeto a partir de una clase, estamos creando una instancia de esa clase. Por lo tanto los objetos no son más que instancias de una clase.

## **PRINCIPIOS DEL MODELO ORIENTADO A OBJETOS**

Los principios más importantes del modelo orientado a objetos son:

abstracción, encapsulación, polimorfismo y herencia, y en un menor grado son tipificación, concurrencia y persistencia. (Booch, 1986).

Si un modelo que se dice orientado a objetos, no contiene alguno de los primeros cuatro elementos, entonces no es orientado a objetos.

### **A. ABSTRACCIÓN**

Mediante la abstracción la mente humana modela la realidad en forma de objetos. Para ello se busca parecidos entre la realidad y la posible implementación de objetos del programa que simulen el funcionamiento de los objetos reales. Significa entender la realidad como objetos con comportamientos bien definidos. Es una descripción simplificada o especificación de un sistema que enfatiza algunos de los detalles o propiedades del sistema, mientras suprime otros. Pero la abstracción humana se gestiona de una manera jerárquica, dividiendo sucesivamente sistemas complejos en conjuntos de subsistemas, para así entender más fácilmente la realidad.

### **B. ENCAPSULACIÓN**

En el proceso de ocultar todos los detalles de un objeto que no contribuyen a sus características esenciales. El encapsulamiento permite a los objetos elegir qué información es publicada y qué información es ocultada al resto de los objetos. Para ello los objetos suelen presentar sus métodos como interfaces públicas y sus atributos como datos privados e inaccesibles desde otros objetos. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas. Esto asegura que otros objetos no pueden cambiar el estado de un objeto de manera inesperada. Con el encapsulado de los datos se consigue que las personas que utilicen un objeto sólo tengan que comprender su interfaz, olvidándose de cómo está implementada.

### **C. MODULARIDAD**

La Modularidad es la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan

independiente como sea posible de la aplicación en sí y de las restantes partes. Por supuesto no todos los módulos son iguales: tomar un programa monolítico y separarlo de forma aleatoria en archivos no es óptimo. Se debe tener en cuenta los conceptos asociados de dependencia, acoplamiento, cohesión, interfaz, encapsulación y abstracción. Una vez identificado lo que es un buen módulo, se puede contemplar la reutilización de un buen módulo como componente. La dependencia a veces se conoce como acoplamiento. Un sistema con muchas dependencias tiene fuerte acoplamiento. Los buenos sistemas tienen débil acoplamiento, porque en ese caso los cambios en una parte del sistema son menos probables de propagarse a través del sistema. Un módulo, de cualquier tamaño y complejidad, es una buena si tiene fuerte cohesión y débil acoplamiento y puede ser factible reutilizarlo en sistemas posteriores, o sustituirlo en el sistema existente.

#### **D. JERARQUÍA**

La Jerarquía es una propiedad que permite la ordenación de las abstracciones. Las dos jerarquías más importantes de un sistema complejo son: estructura de clases (jerarquía "es-un" (is-a): generalización/especialización) y estructura de objetos (jerarquía "parte-de" (part-of): agregación). Las jerarquías de generalización/especialización se conocen como herencia. Básicamente, herencia nos permite definir clases a partir de otras, donde la nueva clase hereda todos los atributos y métodos de la clase antecesora. La herencia puede ser simple o múltiple, una herencia simple tiene como antecesor a una sola clase, a diferencia de la herencia múltiple que tiene más de un antecesor. La agregación es el concepto que permite el agrupamiento físico de estructuras relacionadas lógicamente. Así, un camión se compone de ruedas, motor, sistema de transmisión y chasis; en consecuencia, camión es una agregación, y ruedas, motor, transmisión y chasis son agregados de camión.

#### **E. POLIMORFISMO**

Comportamientos diferentes, asociados a objetos distintos. Es la

habilidad de dos o más clases de responder al mismo mensaje de forma diferente, por ejemplo calcular el perímetro en las clases cuadrado, triangulo, circulo. Por lo que se puede definir el método perímetro cada uno especializado para cada clase. Polimorfismo quiere decir "un objeto y muchas formas". Esta propiedad permite que un objeto presente diferentes comportamientos en función del contexto en que se encuentre. Por ejemplo un método puede presentar diferentes implementaciones en función de los argumentos que recibe, recibir diferentes números de parámetros para realizar una misma operación, y realizar diferentes acciones dependiendo del nivel de abstracción en que sea llamado.

## **F. OTRAS PROPIEDADES**

El modelo objeto ideal no sólo tiene las propiedades anteriormente citadas sino que es conveniente que soporte, además, estas otras propiedades:

- a) Concurrencia (multitarea):** el lenguaje soporta la creación de procesos paralelos independientes del sistema operativo. Esta propiedad simplificará la transportabilidad de un sistema de tiempo real de una plataforma a otra.
- b) Persistencia:** los objetos han de poder permanecer después de la ejecución del programa, su existencia trasciende el tiempo (es decir, el objeto continúa existiendo después de que su creador ha dejado de existir) y/o el espacio (es decir, la localización del objeto se mueve del espacio de dirección en que fue creado).
- c) Genericidad:** las clases parametrizadas (mediante plantillas o unidades genéricas) sirven para soportar un alto grado de reutilización. Estos elementos genéricos se diseñan con parámetros formales, que se instanciarán con parámetros reales, para crear instancias de módulos que se compilan y enlazan, y ejecutan posteriormente.
- d) Manejo de Excepciones:** se deben poder detectar, informar y manejar condiciones excepcionales utilizando construcciones del lenguaje. Esta propiedad añadida al soporte de tolerancia a fallos del software permitirá una estrategia de diseño eficiente.

## RELACIONES ENTRE OBJETOS

Las clases y objetos no pueden existir aislados y en consecuencia existirán relaciones entre ellos, las relaciones entre objetos pueden indicar alguna forma de compartición, así como algún tipo de conexión semántica. Por ejemplo: las margaritas y las rosas son ambas tipos de flores, significando que ambas tienen pétalos coloreados brillantemente, ambas emiten fragancia, etc. (Booch, 1995).

Existen varios tipos de relaciones que pueden unir a los diferentes objetos, pero entre ellas destacan las relaciones de: Generalización y Agregación.

### GENERALIZACIÓN/ ESPECIALIZACIÓN (ES-UN)

Herencia es la propiedad por la cual instancias de una clase hija (sub clase) puede acceder tanto a datos como a comportamientos (métodos) asociados con una clase padre (superclase). La Herencia siempre es transitiva, de modo que una clase puede heredar características de superclases de nivel superior. (Joyanes, 2002).

Gráficamente, la generalización se representa con una línea con punta de flecha vacía.

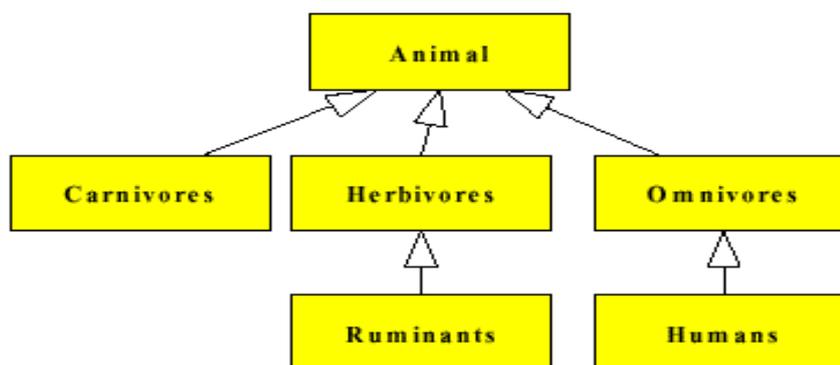


Figura N° 2.8: Representación de Herencia (Joyanes, 2002).

### AGREGACIÓN Y COMPOSICIÓN (TODO/PARTE)

Es una relación que representa a los objetos compuestos, un objeto es compuesto si se compone a su vez de otros objetos. Dos objetos forman un agregado si existe entre ellos una relación todo-parte. (Joyanes, 1996).

La diferencia entre agregación y composición es que mientras que la composición se entiende que dura durante toda la vida del objeto componedor, en la agregación no tiene por qué ser así.

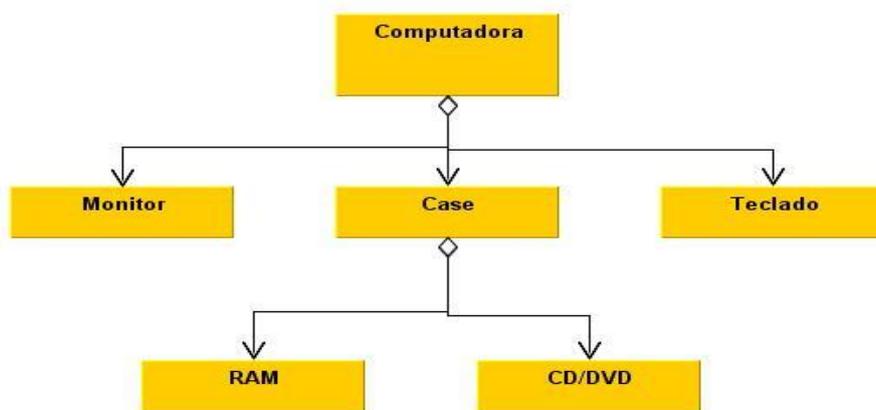


Figura N° 2.9: Representación de la agregación (Joyanes, 2002).

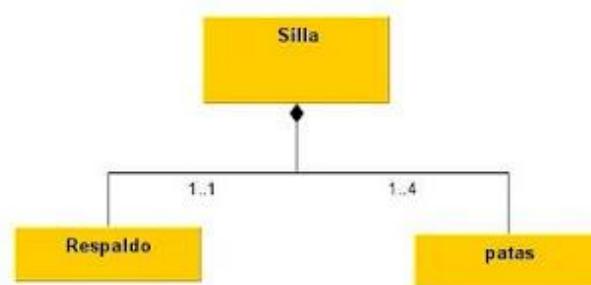


Figura N° 2.10: Representación de la composición (Joyanes, 2002).

### 2.2.7.1 JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. Su creador, James Gosling, lo bautizó como Oak.

Sun deseaba un lenguaje para programar pequeños dispositivos electrónicos (electrodomésticos y otros aparatos electrónicos de consumo. La dificultad de estos dispositivos es que cambian continuamente y para que un programa funcione en el siguiente dispositivo aparecido, hay que reescribir el código. Por eso Sun quería crear un lenguaje independiente del dispositivo.

Se intentaba con este lenguaje paliar uno de los problemas fundamentales del C++: al compilar se produce un fichero ejecutable cuyo código sólo vale para la plataforma en la que se realizó la compilación. La aparición en 1994 del navegador gráfico Mosaic hizo que Internet se popularizara. Sun se dio cuenta de que los logros conseguidos con su proyecto eran perfectamente aplicables a Internet. Básicamente Internet es una gran red mundial que conecta ordenadores con distintos sistemas operativos y distintas arquitecturas. Esta idea hizo que se abandonara el proyecto de desarrollar un lenguaje común para dispositivos electrónicos de consumo y dirigieran sus investigaciones hacia el desarrollo de un lenguaje que permitiera crear aplicaciones que se ejecutaran en cualquier ordenador de Internet.

En 1995 Oak pasa a llamarse Java. Java debe su nombre a un tipo de café. En EEUU se conoce como Java al café, tomarse una taza de Java es tomarse una taza de café. De ahí que el logotipo oficial de Java es una taza humeante de café. Ese año se da a conocer al público y adquiere notoriedad rápidamente, casi desde su lanzamiento. Se comienza a hablar de Java y de sus applets. Un applet es un programa Java que se ejecuta en el contexto de una página web en cualquier ordenador independientemente de su Sistema Operativo y de la arquitectura de su procesador. El entorno de ejecución era relativamente seguro y los principales navegadores web pronto incorporaron la posibilidad de ejecutar applets Java incrustadas en las páginas web.

### **CARACTERÍSTICAS DEL LENGUAJE JAVA**

- a) SENCILLO: Elimina la complejidad de los lenguajes como C y da paso al contexto de los lenguajes modernos orientados a objetos. Aunque la sintaxis de Java es muy similar a C++, elimina algunas de las características más conflictivas de este lenguaje
- b) ORIENTADO A OBJETOS: Es un lenguaje orientado a objetos puro. En Java todo, a excepción de los tipos fundamentales de variables primitivas, son objeto.

- c) **MULTIPLATAFORMA:** Para eliminar la dependencia de la máquina, en Java un programa no se traduce directamente a código ejecutable. Un programa Java (.java) se compila y se obtiene un código llamado bytecode (.class). El bytecode lo interpreta la Máquina Virtual de Java (JVM). También se conoce como JRE (Java Runtime Environment, entorno de ejecución de Java). El JRE o la máquina virtual de Java se distribuye gratuitamente para prácticamente todos los sistemas operativos, lo que significa que un archivo .class se puede ejecutar en cualquier ordenador o máquina que incorpore el JRE.
- d) **ROBUSTO:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.
- e) También implementa los arrays auténticos, en vez de listas enlazadas de punteros, con comprobación de límites, para evitar la posibilidad de sobrescribir o corromper memoria resultado de punteros que señalan a zonas equivocadas. Estas características reducen drásticamente el tiempo empleado en el desarrollo de aplicaciones Java.
- f) **SEGURO:** Al contrario de lo que sucede con C/C++, no se puede acceder a la memoria mediante punteros. Además Java incorpora medidas que evitan que se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no se puede hacer con los recursos críticos de una computadora.
- g) **MULTITAREA:** Soporta múltiples threads, hilos o tareas. Esto quiere decir que puede ejecutar diferentes líneas de código al mismo tiempo tanto si la máquina es multiprocesador como si no lo es.

- h) **DINÁMICO**: En Java no es necesario cargar completamente el programa en memoria sino que las clases compiladas pueden ser cargadas bajo demanda en tiempo de ejecución (dynamic binding).

### **2.2.8 KIT DE DESARROLLO DE SOFTWARE (SDK)**

Un kit de desarrollo de software o SDK (siglas en inglés de Software Development Kit) es generalmente un conjunto de herramientas de desarrollo de software que le permite al programador o desarrollador de software crear aplicaciones para un sistema concreto, por ejemplo ciertos paquetes de software, frameworks, plataformas de hardware, computadoras, videoconsolas, sistemas operativos, etcétera.

Es algo tan sencillo como una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) creada para permitir el uso de cierto lenguaje de programación, o puede, también, incluir hardware sofisticado para comunicarse con un determinado sistema embebido.

#### **2.2.8.1 GRIAULE FINGERPRINT SDK**

El Fingerprint SDK es un Kit de desarrollo innovador para el reconocimiento de huellas dactilares, que le permite integrar datos biométricos en una amplia gama de aplicaciones. Gracias a su apoyo a docenas de lenguajes de programación, la riqueza de código de ejemplos y la documentación en profundidad, empiezan a desarrollar la aplicación.

El Griaule Fingerprint SDK ofrece en dos versiones. (Fingerprint SDK para Windows), soporta múltiples lenguajes de programación a través de Windows DLL o ActiveX (COM). El (Fingerprint SDK para Java) permite el desarrollo para Java multiplataformas. Ejecuta Microsoft Windows y GNU / Linux.

Su objetivo principal es permitir el desarrollo de aplicaciones a medida, tales como los sistemas de autorización, gestión de transacciones, tiempo y

atendimiento, puntos de identificación de ventas y control de acceso físico autenticados mediante la impresión digital.

## **PRINCIPALES FUNCIONALIDADES**

- a) Utilice su lenguaje de programación preferido: Fingerprint SDK soporta múltiples lenguajes de programación como Java, Delphi, Visual Basic, C++, DOTNET, Fox Pro y muchos otros.
- b) Independencia del Lector: Un total de 34 modelos se encuentran entre los más importantes del mercado. Por lo general, los SDKs de los fabricantes solo permiten a utilizar sus propios recursos. La capacidad de soporte del Fingerprint SDK a diversos tipos de lectores de huella digital que permite elegir el más adecuado y, aun después de que el desarrollo y prueba de implementación de su programa, usted puede cambiar el lector, sin cambiar una sola línea de código.
- c) Fácil e intuitivo: Casi todos los SDKs tienen como la interfase con los programas solo DLLs complicadas.
- d) Excelente Velocidad para que coincida con sus huellas dactilares.
- e) Consolidación de la plantilla de huellas dactilares: Mejora el nivel de reconocimiento y elimina la necesidad de utilizar múltiples muestras el mismo dedo, lo que reduce el tamaño de la base de datos y el tiempo identificación.

### **2.2.9 POBLACIÓN Y MUESTRA**

#### **POBLACIÓN**

La población se define como la totalidad del fenómeno a estudiar donde las unidades de población poseen una característica común la cual se estudia y da origen a los datos de la investigación (Tamayo y Tamayo, 1997).

Una población es un conjunto de todos los elementos que estamos estudiando, acerca de los cuales intentamos sacar conclusiones. (Leviny y Rubin, 1996).

La población es el conjunto de todos los casos que concuerdan con una serie de

especificaciones, podemos decir que la población es la totalidad del fenómeno a estudiar, en donde las unidades de población posee una característica común la cual estudia y da origen a los datos. (Hernández, 2000).

Población es el conjunto total de individuos, objetos o medidas que poseen algunas características comunes observables en un lugar y en un momento determinado. Cuando se vaya a llevar a cabo alguna investigación debe de tenerse en cuenta algunas características esenciales al seleccionarse la población bajo estudio. Entre éstas tenemos:

- a) Homogeneidad, que todos los miembros de la población tengan las mismas características según las variables que se vayan a considerar en el estudio o investigación.
- b) Tiempo, se refiere al período de tiempo donde se ubicaría la población de interés. Determinar si el estudio es del momento presente o si se va a estudiar a una población de cinco años atrás o si se van a entrevistar personas de diferentes generaciones.
- c) Espacio, se refiere al lugar donde se ubica la población de interés. Un estudio no puede ser muy abarcador y por falta de tiempo y recursos hay que limitarlo a un área o comunidad en específico.
- d) Cantidad, se refiere al tamaño de la población. El tamaño de la población es sumamente importante porque ello determina o afecta al tamaño de la muestra que se vaya a seleccionar, además que la falta de recursos y tiempo también nos limita la extensión de la población que se vaya a investigar.

## **MUESTRA**

La muestra se define como un subgrupo de la población, para delimitar las características de la población. (Hernández, 2000).

Como afirma Acevedo (1984) se define a la muestra como una población o sea, un número de individuos, un objeto de los cuales es un elemento del universo o población, es decir, un conjunto de la población con la que se está trabajando

por lo cual esta investigación se circunscribe a la cantidad de casos que se gestionan.

Como afirma Muñoz (1998), se define a la muestra como un instrumento que supone la obtención de datos de todas las unidades del universo acerca de las cuestiones que constituyen el objeto del censo.

Para calcular el tamaño de la muestra con una población finita, se utiliza la siguiente fórmula:

$$n = \frac{N * Z_{\alpha}^2 * p * q}{d^2 * (N - 1) + Z_{\alpha}^2 * p * q}$$

Dónde:

n = el tamaño de la muestra

N = tamaño de la población

Z $\alpha$  = Valor obtenido mediante niveles de confianza. Es un valor constante que, si no se tiene su valor, se lo toma en relación al 95% de confianza equivale a 1,96 (como más usual) o en relación al 99% de confianza equivale 2,58, valor que queda a criterio del investigador

p = proporción esperada

q = 1 - p

d = precisión.

## 2.2.10 MUESTREO

### MUESTREO POR CONVENIENCIA

El muestreo por conveniencia es un método de muestreo no probabilístico. Consiste en seleccionar a los individuos que convienen al investigador para la muestra. Esta conveniencia se produce porque al investigador le resulta más sencillo examinar a estos sujetos, ya sea por proximidad geográfica, por ser sus amigos, etc. El método del muestreo por conveniencia permite seleccionar una muestra con muchísima facilidad. Suele utilizarse en estudios iniciales para comprobar si se cumplen las hipótesis que se plantea el investigador. Una vez

realizado el estudio, si se comprueba que los resultados son favorables a sus predicciones, ya se puede plantear la posibilidad de hacer el estudio con muestras probabilísticas para generalizar el resultado.

Este tipo de muestreo se caracteriza por un esfuerzo deliberado de obtener muestras "representativas" mediante la inclusión en la muestra de grupos supuestamente típicos. Es muy frecuente su utilización en sondeos preelectorales de zonas que en anteriores votaciones han marcado tendencias de voto. También puede ser que el investigador seleccione directa e intencionadamente los individuos de la población. El caso más frecuente de este procedimiento es el utilizar como muestra los individuos a los que se tiene fácil acceso (los profesores de universidad emplean con mucha frecuencia a sus propios alumnos).

### **MUESTREO ALEATORIO ESTRATIFICADO**

El muestreo aleatorio estratificado, consiste en considerar categorías típicas diferentes entre sí (estratos) que poseen gran homogeneidad respecto a alguna característica (se puede estratificar, por ejemplo, según la profesión, el municipio de residencia, el sexo, el estado civil, etc.). Lo que se pretende con este tipo de muestreo es asegurarse de que todos los estratos de interés estarán representados adecuadamente en la muestra. Cada estrato funciona independientemente, pudiendo aplicarse dentro de ellos el muestreo aleatorio simple o el estratificado para elegir los elementos concretos que formarán parte de la muestra. En ocasiones las dificultades que plantean son demasiado grandes, pues exige un conocimiento detallado de la población. (Tamaño geográfico, sexos, edades,...).

Este es uno de los procedimientos más empleados en la formación de muestras. Consiste, básicamente, en dividir a la población en diferentes segmentos denominados estratos, formados por elementos lo más homogéneamente posibles entre sí. Los estratos se forman atendiendo a diferentes variables, como pueden ser: sexo, edad, nivel de renta, etc. De cada uno de los estratos

formados se procede a elegir una muestra mediante alguno de los procedimientos anteriores. La precisión de la encuesta aumenta, generalmente, a medida que se incrementa el número de elementos a elegir de cada estrato.

## **CAPÍTULO III**

### **METODOLOGÍA DE LA INVESTIGACIÓN**

#### **3.1 TIPO DE INVESTIGACIÓN**

Como afirma Bernal (2006), la investigación descriptiva es uno de los tipos o procedimientos investigativos más populares y utilizados por los principiantes en la actividad investigativa. Los trabajos de grado, en los pregrados y en muchas de las maestrías, son estudios de carácter eminentemente descriptivo. En tales estudios se muestran, narran, reseñan o identifican hechos, situaciones, rasgos, características de un objeto de estudio, o se diseñan productos, modelos, prototipos, guías, etcétera. Pero no dan explicaciones o razones del porqué de las situaciones, los hechos, los fenómenos, etcétera. Según la definición de investigación descriptiva y el estudio, se está creando un prototipo de software para la identificación de actores en la Universidad Nacional de San Cristóbal de Huamanga, mediante el control biométrico para procesos críticos de gestión universitaria; por esta consideración el tipo de investigación es descriptiva.

#### **3.2 NIVEL DE INVESTIGACIÓN**

Como afirma Carrasco (2006), la investigación aplicada se distingue por tener propósitos prácticos inmediatos definidos es decir, se investiga para actuar, transformar, modificar o producir cambios en un determinado sector de la realidad. Para realizar investigaciones aplicadas es muy importante contar con el aporte de las teorías científicas, que son producidas por la investigación básica y sustantiva. De acuerdo a este concepto el prototipo de software de identificación de un actor mediante el control biométrico permite producir cambios en los procesos de la identificación de los actores; por esta consideración el nivel de investigación es aplicada.

### **3.3 DISEÑO DE LA INVESTIGACIÓN**

Como afirma Aratoma (2007), una investigación no experimental es “toda investigación en las que son imposibles manipular variables o asignar aleatoriamente a los sujetos o las condiciones”. En el presente estudio, la primera variable, procesos críticos de gestión universitaria no es intervenida a fin de variarla, asimismo, la segunda variable que son los actores; únicamente nos limitamos a identificarla y realizar la automatización de ésta; se recopila la información en único momento, por las consideraciones expuestas, el estudio es no experimental de diseño transeccional descriptivo.

### **3.4 POBLACIÓN Y MUESTRA**

**POBLACIÓN.-** La población estuvo compuesta por 125 procesos de gestión universitaria que realizan los actores de la Universidad Nacional San Cristóbal de Huamanga en el año 2016.

**MUESTRA.-** Se hizo el muestreo por conveniencia de los procesos de gestión universitaria, compuesta por 19 procesos críticos automatizables que realizan los actores. Asimismo, se realizó el muestreo aleatorio estratificado de los actores de la Universidad con un 95% de confianza y un 5% de error, siendo las muestras de 236 docentes, 373 estudiantes, 365 postulantes y 223 ingresantes.

### **3.5 VARIABLES E INDICADORES**

#### **DEFINICIÓN CONCEPTUAL DE LAS VARIABLES**

#### **PRIMERA VARIABLE**

**Procesos críticos de gestión universitaria.-** Son aquellos sin los que sería difícil garantizar la calidad en el cumplimiento del servicio, es decir, son procesos significativos vinculados a cada tipo de organización.

#### **INDICADORES DE LA PRIMERA VARIABLE**

**Gestión académica.-** son los procesos orientados a mejorar la calidad

educativa y mejorar los procesos pedagógicos, con el fin de responder a las necesidades académicas actuales.

**Gestión administrativa.-** Es el proceso de diseñar y mantener un entorno en el que, trabajando en grupos, los individuos cumplen eficientemente objetivos específicos.

## **SEGUNDA VARIABLE**

**Actor.-** Un actor es una agrupación uniforme de personas, sistemas o máquinas que interactúan con el sistema que estamos construyendo.

## **INDICADORES DE LA SEGUNDA VARIABLE**

**Estudiante.-** Es la palabra que permite referirse a quienes se dedican a la aprehensión, puesta en práctica y lectura de conocimientos sobre alguna ciencia, disciplina o arte.

**Docente universitario.-** Un docente es aquel individuo que se dedica a enseñar o que realiza acciones referentes a la enseñanza.

**Postulante.-** Es la persona que, habiendo completado sus estudios secundarios, o que estando por completarlos en el año previo al inicio del año académico en la universidad, desea seguir sus estudios superiores en una Universidad.

## **DEFINICIÓN OPERACIONAL DE LAS VARIABLES**

### **PRIMERA VARIABLE**

X: Procesos críticos de gestión universitaria.

### **INDICADORES DE LA PRIMERA VARIABLE**

X1: Gestión académica.

X2: Gestión administrativa.

## **SEGUNDA VARIABLE**

Y: Actor

## **INDICADORES DE LA SEGUNDA VARIABLE**

Y1: Estudiante

Y2: Docente universitario

Y3: Postulante

## **OPERACIONALIZACIÓN DE LAS VARIABLES**

Se ha operacionalizado las variables para definir los ítems que permiten obtener la información sobre las variables de investigación, detallados en el Anexo A.

### **3.6 TÉCNICAS E INSTRUMENTOS PARA EL LEVANTAMIENTO DE INFORMACIÓN**

#### **TÉCNICAS PARA RECOLECTAR LA INFORMACIÓN**

Las técnicas que se usaron para levantar la información, son el análisis documental, entrevista y encuesta.

**Análisis documental.-** Para el análisis documental, se revisaron los siguientes documentos: reglamento universitario, el estatuto, el Manual de Organización y Función de la UNSCH, el Reglamento de Organización y Función de la UNSCH, el currículo de estudios, el Texto Único de Procedimientos Administrativos y el prospecto de admisión.

**Entrevista.-** Se utilizó la técnica de entrevista para recolectar información sobre el proceso de admisión. La entrevista se realizó al presidente de la comisión de admisión.

**Encuesta.-** Se utilizó la técnica de encuesta a los docentes universitarios, estudiantes, postulantes e ingresantes.

## **INSTRUMENTOS PARA RECOLECTAR LA INFORMACIÓN**

**Ficha de análisis documental.-** Se usó el anexo B, instrumento que nos permite identificar los procesos críticos automatizables.

**Guía para entrevista.-** Se usó el anexo C, instrumento que nos permite realizar la entrevista al presidente de la comisión de admisión, para conocer las medidas de control durante los procesos del examen de admisión.

**Cuestionario.-** Se usó el anexo D, como instrumento para recolectar información relacionada al control de asistencia de los docentes a sus diversas actividades, a los estudiantes que usan los servicios de la universidad y a los postulantes e ingresantes durante los procesos de admisión e ingreso.

### **3.6.1 HERRAMIENTAS PARA EL TRATAMIENTO DE LA INFORMACIÓN**

Las herramientas tecnológicas que se utilizaron, para desarrollar el software de control biométrico, son en gran parte software libre, debido a que la universidad no cuenta con un software de este tipo, y no se desea hacer gastos en licencia, ya que existen limitaciones económicas y de idiosincrasia que se generan en el entorno. Las herramientas fueron seleccionadas en función al marco teórico, mencionadas en el capítulo II, sección 2.2.6.1, 2.2.7.1 y 2.2.8.1.

<b>SOFTWARE</b>	<b>FABRICANTE</b>	<b>SERVICIO</b>
WINDOWS 7	Producida por Microsoft Corporation	Windows Seven
JAVA	Desarrollado por Sun Microsystems	Java es un lenguaje de programación orientado a objetos que posee un sistema que interpreta y ejecuta los archivos para ser compilados en tiempo real
NETBEANS IDE	Fundado por SunMicroSystems, 2000	Una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación.

MySQL	Community MySQL	MySQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia GPL.
SDK	Desarrollado por Griaule	Es un kit de desarrollo para Java, que nos permite interactuar con dispositivos lectores de huella conectados a nuestra PC.

Tabla N° 3.1: Herramientas tecnológicas para el tratamiento de datos.

### 3.6.2 TÉCNICAS PARA APLICAR LA METODOLOGÍA XP

Según el capítulo 2, la sección 2.2.5, el proceso, que se considera para desarrollar el software con la metodología XP, es el que se muestra a continuación.

ARTEFACTO	ACTIVIDAD	RESPONSABLES
Historia de usuario	Describir brevemente la historia de usuario con la regla del negocio (lo que el sistema debe hacer)	Cliente
Plan de Entregas	Estimar esfuerzo (semana) para desarrollar la historia de usuario	Cliente Programador Entrenador
Plan de iteración	Identificar las historias de usuario que se van a desarrollar en una iteración específica	Programador

Tabla N° 3.2: Fase de planificación.

ARTEFACTO	ACTIVIDAD	RESPONSABLES
Metáfora del Sistema	Definir el nombre de las clases y de los métodos.	Programador
Tarjetas CRC	Diseñar para una tarea de ingeniería de forma simple. Rediseñar por falla de prueba de aceptación una tarea. Identificar responsabilidades. Identificar atributos.	Cliente Programador
Soluciones puntuales	Remover la redundancia. Eliminar las funcionalidades no necesarias. Rejuvenecer los diseños obsoletos.	Programador

Tabla N° 3.3: Fase de diseño.

ARTEFACTO	ACTIVIDAD	RESPONSABLES
Código Fuente	Codificar una tarea de ingeniería. Hacer refactoring. Mover programadores.	Programador Supervisor
Prueba unitaria	Escribir código fuente para una prueba unitaria, usando una herramienta. Ejecutar el módulo de cada prueba unitaria. Modificar código fuente si la prueba unitaria muestra resultado incorrecto.	Programador
Integración	Integrar las tareas para una historia de usuario. Mantener sistema integrado todo el tiempo. Integrar continuamente al concluir las tareas de una historia de usuario. Verificar que las pruebas de integración pasan al 100%.	Programador

Tabla N° 3.4: Fase de desarrollo.

ARTEFACTO	ACTIVIDAD	RESPONSABLES
Pruebas de aceptación	Crear pruebas de aceptación a partir de las historias de usuario. Correr la última versión de una iteración Utilizar los casos de prueba de aceptación	Cliente

Tabla N° 3.5: Fase de pruebas.

## CAPÍTULO IV

### ANÁLISIS Y RESULTADOS DE LA INVESTIGACIÓN

#### 4.1. ARTEFACTOS DEL SOFTWARE APLICANDO EL PROCESO XP

##### 4.1.1 FASE DE EXPLORACIÓN

Según el procedimiento desarrollado en la tabla 3.2 para el proceso XP, descrito en el capítulo III, fase de planificación. Obtenemos las historias de usuario, arquitectura técnica inicial y el plan de alto nivel.

Nº	Historia de usuario	Descripción
1	Registrar y mantener datos de postulante	El usuario (Comisión de admisión), podrá registrar y mantener los datos personales, fotografía y datos de admisión de un postulante.
2	Mantener huella de postulante	El usuario (Comisión de admisión), podrá consultar y eliminar las huellas registradas de un postulante.
3	Consultar datos de postulante	El usuario (Comisión de admisión), podrá consultar los datos de un postulante, a partir del número de su DNI.
4	Registrar y mantener proceso de admisión	El usuario (Comisión de admisión), podrá registrar y crear en el sistema, el nombre del proceso de admisión, para que se pueda visualizar en el sistema.
5	Verificar identidad de postulante	El usuario (Comisión de admisión), podrá realizar el control de acceso al examen de admisión, por medio de la verificación de la identidad de los postulantes.
6	Emisión de certificado de ingreso	El usuario (Comisión de admisión), podrá realizar la emisión de los certificados de ingreso, a partir de la validación de la identidad del ingresante.
7	Registro y mantenimiento de datos del docente	El usuario (Personal administrativo), podrá registrar y mantener los datos personales de un docente.
8	Controlar docente	El usuario (docente), podrá registrar su asistencia al dictado de clases y sesiones de departamento.
9	Emisión de acta de notas	El usuario (docente) podrá elaborar el registro de notas y la respectiva emisión de las actas de notas, mediante la identificación del docente.
10	Verificar identidad de comensal	El usuario (Personal de bienestar universitario), podrá validar la identidad de los estudiantes usuarios del servicio de comedor.

11	Verificar identidad de usuario de biblioteca	El usuario (Personal de la Biblioteca), podrá validar la identidad de los estudiantes usuarios del servicio de Biblioteca.
----	--	--

Tabla Nº 4.1: Historias de usuario

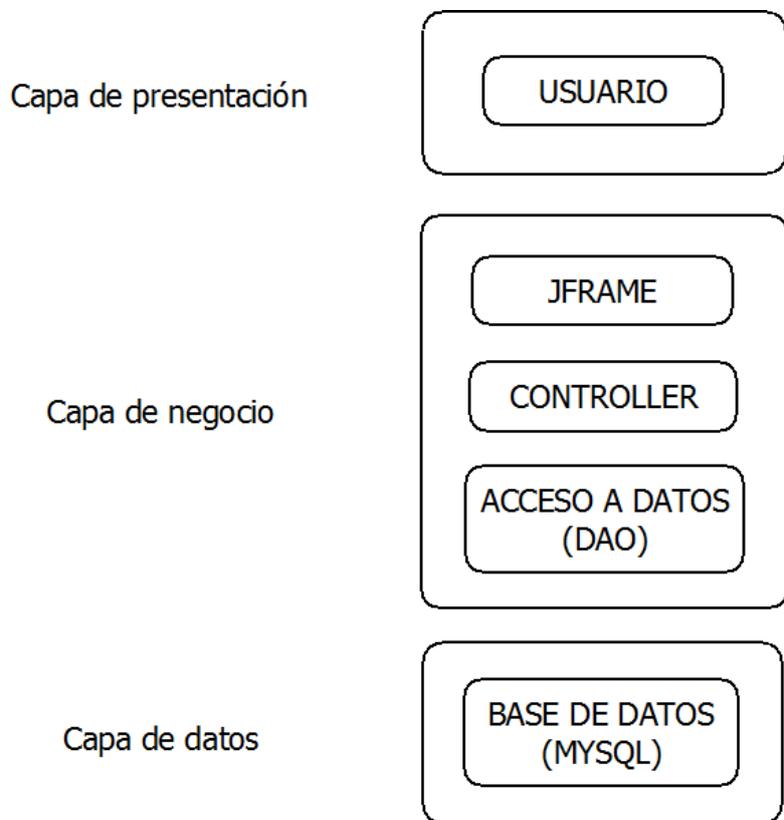


Figura Nº 4.1: Arquitectura técnica inicial

Nº	Historias de Usuario	Esfuerzo (Días)
1	Registrar y mantener datos de postulante	6
2	Mantener huella de postulante	2
3	Consultar datos de postulante	2
4	Registrar y mantener proceso de admisión	2
5	Verificar identidad de postulante	4
6	Emisión de certificado de ingreso	4
7	Registro y mantenimiento de datos del docente	3
8	Controlar docente	3
9	Emisión de acta de notas	4
10	Verificar identidad de comensal	3
11	Verificar identidad de usuario de biblioteca	3

Tabla Nº 4.2: Plan de alto nivel

#### 4.1.2 FASE DE PLANIFICACIÓN

Aplicamos la técnica para la fase de planificación obteniendo los artefactos; historias de usuario en detalle y el plan de versión (primera iteración).

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 1	<b>USUARIO:</b> Comisión de admisión
<b>NOMBRE DE HISTORIA:</b> Registrar y mantener datos de postulante	
<b>PRIORIDAD EN NEGOCIO:</b> Alto	<b>RIESGO EN DESARROLLO:</b> Alto
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI del postulante y realiza la búsqueda de los datos personales del postulante los cuales puede modificar; y en caso no se encuentren los datos del postulante, se ingresarán todos los datos solicitados para registrar el nuevo postulante, además podrá cargar la fotografía del postulante, para finalizar con el registro se hace clic en el botón Guardar.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.3: Historia de usuario. Registrar y mantener datos del postulante.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 2	<b>USUARIO:</b> Comisión de admisión
<b>NOMBRE DE HISTORIA:</b> Mantener huella de postulante	
<b>PRIORIDAD EN NEGOCIO:</b> Alto	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI de un postulante ya registrado y realiza la búsqueda de los datos personales del postulante. Para realizar consulta de las huellas registradas del postulante, hace clic en el botón Consultar huella, el sistema muestra la ventana de Consulta de huellas registradas, en la que podrá visualizar las huellas registradas al postulante. Y si el usuario desea eliminar una de las huellas registradas, debe seleccionar una de las huellas consultadas y hacer clic en el botón Eliminar huella.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.4: Historia de usuario. Mantener huella de postulante.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 3	<b>USUARIO:</b> Comisión de admisión
<b>NOMBRE DE HISTORIA:</b> Consultar datos de postulante	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Buscar postulante del menú Postulante; el sistema muestra la interfaz de Consulta de datos del postulante, el usuario ingresa un DNI y luego hace clic en el botón Buscar, el sistema mostrará los datos del postulante y en caso el DNI ingresado no genere datos, se mostrará un mensaje indicando que el DNI ingresado no está asociado a un postulante.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.5: Historia de usuario. Consultar datos del postulante.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 4	<b>USUARIO:</b> Comisión de admisión
<b>NOMBRE DE HISTORIA:</b> Registrar y mantener proceso de admisión	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Admisión del menú Configuraciones, y el sistema muestra la ventana Configuración del proceso de admisión. Si el usuario desea cambiar el proceso de selección, debe seleccionar el proceso de admisión y hace clic en el botón Guardar. Pero si desea agregar un nuevo proceso de admisión, debe hacer clic en el botón Agregar proceso, el sistema mostrará una ventana con los datos del nuevo proceso, el usuario llenar el formulario y hace clic en Guardar y el sistema creará el registro del nuevo proceso de admisión.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.6: Historia de usuario. Registrar y mantener proceso de admisión.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 5	<b>USUARIO:</b> Comisión de admisión
<b>NOMBRE DE HISTORIA:</b> Verificar identidad de postulante	
<b>PRIORIDAD EN NEGOCIO:</b> Alto	<b>RIESGO EN DESARROLLO:</b> Alto
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Verificar postulante del menú Postulante; el sistema muestra la interfaz de Verificación de postulante, el usuario ingresa el DNI del postulante a verificar la identidad, luego el postulante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del postulante, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del postulante y registrando en la base de datos la fecha y hora de acceso al examen; caso contrario se mostrará un mensaje rechazando la identidad de dicho postulante, registrando la fecha y hora de rechazo.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.7: Historia de usuario. Verificar identidad del postulante.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 6	<b>USUARIO:</b> Comisión de admisión
<b>NOMBRE DE HISTORIA:</b> Emisión de certificado de ingreso	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Certificado de ingreso del menú Ingresante; el sistema muestra la interfaz de Emisión de certificados de ingreso, el usuario ingresa el DNI del ingresante a verificar la identidad, luego el ingresante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del ingresante, en caso la identidad sea verificada se mostrará su certificado de ingreso, registrando en la base de datos la fecha y hora de emisión del certificado; caso contrario se mostrará un mensaje rechazando la identidad de dicho ingresante, registrando la fecha y hora de rechazo.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.8: Historia de usuario. Emisión de certificado de ingreso.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 7	<b>USUARIO:</b> Personal administrativo
<b>NOMBRE DE HISTORIA:</b> Registro y mantenimiento de datos del docente	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Registrar docente del menú Docente; el sistema muestra la interfaz de Registro de datos del docente. El usuario puede hacer la búsqueda de los datos del docente ingresando el DNI del docente, en caso el docente no esté registrado, el usuario ingresa los datos del docente, luego hace clic en el botón Guardar, y si el docente ya existe se puede realizar la modificación de datos en casos sea necesario. Para registrar la huella del docente en la misma ventana, se debe elegir el dedo del cual se registrará la huella, el docente colocará el dedo sobre la superficie del lector de huella digital, y el sistema capturará la huella, luego hace clic en el botón Registrar huella, si desea volver a tomar la huella debe volver a poner el dedo sobre el lector y se volverá a tomar la huella, luego hacer clic en Registrar huella, el sistema preguntará si se desea actualizar la huella, y finalmente para guardar los cambios se hace clic en el botón Guardar.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla N° 4.9: Historia de usuario. Registro y mantenimiento de datos del docente.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 8	<b>USUARIO:</b> Docente
<b>NOMBRE DE HISTORIA:</b> Controlar docente	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El docente, hace clic en la opción Asistencia del menú Docente; el sistema muestra la interfaz de Control de asistencia de docente, en la interfaz se muestra un marcador de hora en el que el docente podrá ingresar su DNI, luego el docente coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del docente, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del docente y registrando en la base de datos la fecha y hora de marcado de asistencia; caso contrario se mostrará un mensaje rechazando la identidad de dicho docente y el docente podrá repetir la acción.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla N° 4.10: Historia de usuario. Controlar docente.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 9	<b>USUARIO:</b> Docente
<b>NOMBRE DE HISTORIA:</b> Emisión de acta de notas	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El docente, hace clic en la opción Acta de notas del menú Docente; el sistema muestra la interfaz de Verificación de docente, el docente ingresa su DNI la identidad, luego el docente coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del docente, en caso la identidad sea rechazada se mostrará un mensaje indicando que la verificación de identidad no se realizó con éxito, y si la identidad del docente es verificada se mostrará la ventana de Emisión de acta de notas, en el que el docente podrá seleccionar un	

curso que dicta y podrá realizar el registro de notas finales por alumnos matriculados en el curso seleccionado, una vez que el docente termine de registrar las notas finales, el docente podrá guardar el registro de notas haciendo clic en el botón Guardar, y en caso desee emitir el acta de notas deberá hacer clic en el botón Emitir acta, y el sistema le mostrará una ventana de confirmación en el que el docente deberá colocar sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realizará la verificación de la identidad del docente, en caso la identidad sea confirmada se emitirá el acta de notas del curso; caso contrario no se emitirá ningún acta de notas.

**OBSERVACIONES:** Ninguna

Tabla Nº 4.11: Historia de usuario. Emisión de acta de notas.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 10	<b>USUARIO:</b> Personal de bienestar universitario
<b>NOMBRE DE HISTORIA:</b> Verificar identidad de comensal	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Verificar comensal del menú Bienestar universitario; el sistema muestra la interfaz de Verificación de comensal, el usuario ingresa el DNI del estudiante a verificar la identidad, luego el estudiante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del estudiante, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del estudiante y registrando en la base de datos la fecha y hora de acceso del estudiante al comedor; caso contrario se mostrará un mensaje rechazando la identidad de dicho estudiante, registrando la fecha y hora de rechazo.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.12: Historia de usuario. Verificar identidad del comensal.

<b>HISTORIAS DE USUARIO</b>	
<b>NÚMERO:</b> 11	<b>USUARIO:</b> Personal de Biblioteca
<b>NOMBRE DE HISTORIA:</b> Verificar identidad de usuario de biblioteca	
<b>PRIORIDAD EN NEGOCIO:</b> Medio	<b>RIESGO EN DESARROLLO:</b> Medio
<b>PUNTOS ESTIMADOS:</b> 1	<b>ITERACIÓN ASIGNADA:</b> 1
<b>PROGRAMADOR RESPONSABLE:</b> Juan Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Verificar usuario de biblioteca del menú Biblioteca universitaria; el sistema muestra la interfaz de Verificación de usuario de biblioteca, el usuario ingresa el DNI del estudiante a verificar la identidad, luego el estudiante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del estudiante, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del estudiante y registrando en la base de datos la fecha y hora de solicitud de préstamo de material bibliográfico a domicilio; caso contrario se mostrará un mensaje rechazando la identidad de dicho estudiante, registrando la fecha y hora de rechazo.	
<b>OBSERVACIONES:</b> Ninguna	

Tabla Nº 4.13: Historia de usuario. Verificar identidad del usuario de biblioteca.

Nº	Historias de Usuario	Prioridad	Riesgo	Esfuerzo (Días)	Iteración
1	Registrar y mantener datos de postulante	Alto	Alto	6	2
2	Mantener huella de postulante	Medio	Medio	2	1
3	Consultar datos de postulante	Medio	Medio	2	1
4	Registrar y mantener proceso de admisión	Medio	Medio	2	1
5	Verificar identidad de postulante	Alto	Alto	4	2
6	Emisión de certificado de ingreso	Alto	Alto	4	2
7	Registro y mantenimiento de datos del docente	Alto	Alto	3	2
8	Controlar docente	Alto	Alto	3	2
9	Emisión de acta de notas	Alto	Alto	4	2
10	Verificar identidad de comensal	Alto	Alto	3	2
11	Verificar identidad de usuario de biblioteca	Alto	Alto	3	2

Tabla Nº 4.14: Plan de versión.

### 4.1.3 FASE DE ITERACIÓN

La fase de iteración permite obtener los entregables; arquitectura técnica, tareas de ingeniería, plan de iteración, casos de prueba de aceptación, GUI, tarjetas CRC, base de datos física, código fuente para clases entidad, pruebas unitarias, código fuente para tarea de ingeniería, reporte de pruebas unitarias, reporte de pruebas de integración y de aceptación.

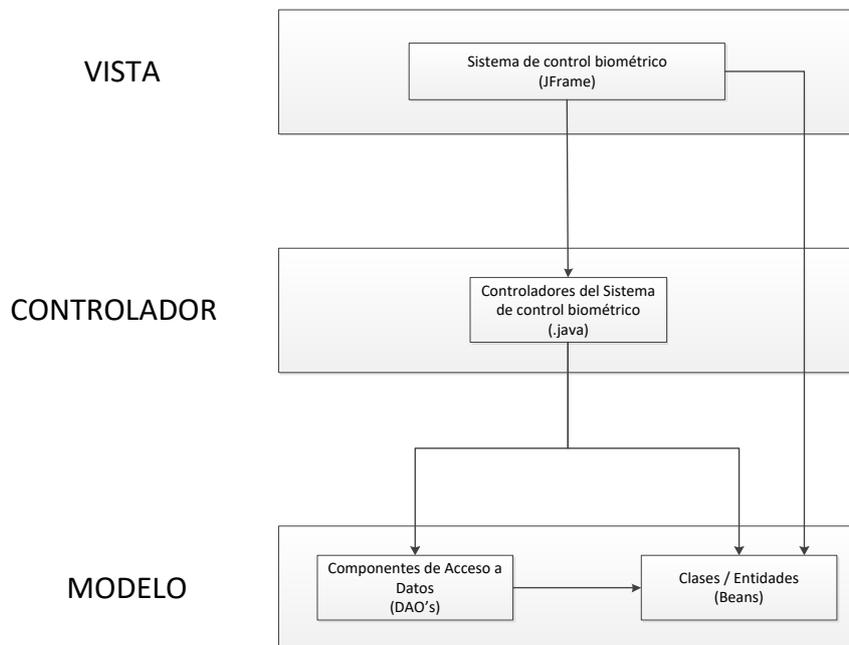


Figura Nº 4.2: Diagrama de componentes.

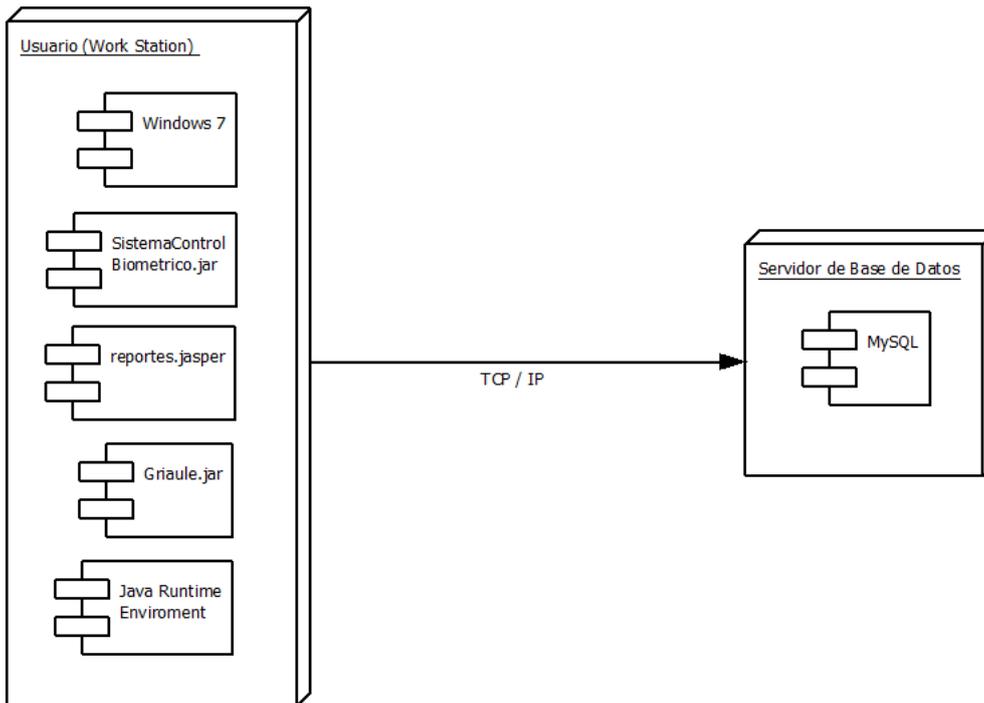


Figura N° 4.3: Diagrama de despliegue.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 1
<b>Nombre de tarea:</b> Registrar datos del postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 6
<b>Fecha de inicio:</b> 01/06/1016	<b>Fecha de fin:</b> 08/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI del postulante y realiza la búsqueda de los datos personales del postulante, en caso no se encuentren los datos del postulante, se ingresarán todos los datos solicitados, para grabar los cambios, se hace clic en el botón Guardar.	

Tabla N° 4.15: Tareas de ingeniería. Registrar postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 2	<b>Número de historia de usuario:</b> 1
<b>Nombre de tarea:</b> Registrar huellas de postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 6
<b>Fecha de inicio:</b> 01/06/1016	<b>Fecha de fin:</b> 08/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI del postulante ya existente y selecciona en el combo el dedo del cual desea capturar la huella digital, el postulante coloca su dedo sobre la superficie del lector de huella digital, y el sistema captura la huella, mostrando la imagen de la huella en la interfaz. El usuario hace clic en el botón Registrar huella, y si ya no desea registrar más huellas, finaliza el registro de huella haciendo clic en el botón Guardar.	

Tabla N° 4.16: Tareas de ingeniería. Registrar huella del postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 3	<b>Número de historia de usuario:</b> 1
<b>Nombre de tarea:</b> Actualizar datos de postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 6
<b>Fecha de inicio:</b> 01/06/1016	<b>Fecha de fin:</b> 08/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI de un postulante ya registrado y realiza la búsqueda de los datos personales del postulante, paso seguido el usuario modifica los datos personales del postulante y para finalizar con la actualización de los datos, hace clic en el botón Guardar.	

Tabla N° 4.17: Tareas de ingeniería. Actualizar datos del postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 4	<b>Número de historia de usuario:</b> 1
<b>Nombre de tarea:</b> Cargar fotografía de postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 6
<b>Fecha de inicio:</b> 01/06/1016	<b>Fecha de fin:</b> 08/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI de un postulante ya registrado y realiza la búsqueda de los datos personales del postulante, paso seguido el usuario hace clic en el botón Cargar imagen y el sistema muestra un explorador de archivos, en el cual debe elegir la fotografía del postulante una vez seleccionada la fotografía, hace clic en el botón Guardar.	

Tabla N° 4.18: Tareas de ingeniería. Cargar fotografía de postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 2
<b>Nombre de tarea:</b> Consultar registro de huella de postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 09/06/1016	<b>Fecha de fin:</b> 10/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI de un postulante ya registrado y realiza la búsqueda de los datos personales del postulante, El usuario hace clic en el botón Consultar huella, el sistema muestra la ventana de Consulta de huellas registradas, en la que podrá visualizar las huellas registradas al postulante.	

Tabla N° 4.19: Tareas de ingeniería. Consultar registro de huella de postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 2	<b>Número de historia de usuario:</b> 2
<b>Nombre de tarea:</b> Eliminar registro de huella de postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 09/06/1016	<b>Fecha de fin:</b> 10/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar Huella de postulante, del menú Postulante, el sistema muestra la interfaz de Registro de huella de postulante, el usuario ingresa el DNI de un postulante ya registrado y realiza la búsqueda de los datos personales del	

postulante. Si el usuario desea eliminar una de las huellas registradas, deberá hacer clic en el botón Consulta de huellas registradas, de la lista de las huellas consultadas seleccionar una y hacer clic en el botón Eliminar huella.

Tabla N° 4.20: Tareas de ingeniería. Eliminar registro de huella de postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 3
<b>Nombre de tarea:</b> Buscar datos de postulante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 13/06/2016	<b>Fecha de fin:</b> 14/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Buscar postulante del menú Postulante; el sistema muestra la interfaz de Consulta de datos del postulante, el usuario ingresa un DNI y luego hace clic en el botón Buscar, el sistema mostrará los datos del postulante y en caso el DNI ingresado no genere datos, se mostrará un mensaje indicando que el DNI ingresado no está asociado a un postulante.	

Tabla N° 4.21: Tareas de ingeniería. Buscar datos de postulante.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 4
<b>Nombre de tarea:</b> Registrar proceso de admisión	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 13/06/1016	<b>Fecha de fin:</b> 14/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Admisión del menú Configuraciones, y el sistema muestra la ventana Configuración del proceso de admisión. Si el usuario desea agregar un nuevo proceso de admisión, debe hacer clic en el botón Agregar proceso, el sistema mostrará una ventana con los datos del nuevo proceso, el usuario llenar el formulario y hace clic en Guardar y el sistema creará el registro del nuevo proceso de admisión.	

Tabla N° 4.22: Tareas de ingeniería. Registrar proceso de admisión.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 2	<b>Número de historia de usuario:</b> 4
<b>Nombre de tarea:</b> Definir el proceso de admisión	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2
<b>Fecha de inicio:</b> 15/06/1016	<b>Fecha de fin:</b> 16/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Admisión del menú Configuraciones, y el sistema muestra la ventana Configuración del proceso de admisión, el usuario selecciona el proceso de admisión y hace clic en el botón Guardar.	

Tabla N° 4.23: Tareas de ingeniería. Definir el proceso de admisión.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 5
<b>Nombre de tarea:</b> Verificar identidad de postulante mediante su huella al momento de acceso al Examen de Admisión	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Fecha de inicio:</b> 17/06/1016	<b>Fecha de fin:</b> 22/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>DESCRIPCIÓN:</b>	

El usuario, hace clic en la opción Verificar postulante del menú Postulante; el sistema muestra la interfaz de Verificación de postulante, el usuario ingresa el DNI del postulante a verificar la identidad, luego el postulante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del postulante, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del postulante y registrando en la base de datos la fecha y hora de acceso al examen; caso contrario se mostrará un mensaje rechazando la identidad de dicho postulante, registrando la fecha y hora de rechazo.

Tabla N° 4.24: Tareas de ingeniería. Verificar identidad de postulante mediante su huella al momento de acceso al Examen de Admisión.

<b>TAREAS DE INGENIERÍA</b>	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 6
<b>Nombre de tarea:</b> Emitir certificado de ingreso mediante verificación de identidad de ingresante	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 23/06/1016	<b>Fecha de fin:</b> 28/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>DESCRIPCIÓN:</b> El usuario, hace clic en la opción Certificado de ingreso del menú Ingresante; el sistema muestra la interfaz de Emisión de certificados de ingreso, el usuario ingresa el DNI del ingresante a verificar la identidad, luego el ingresante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del ingresante, en caso la identidad sea verificada se mostrará su certificado de ingreso, registrando en la base de datos la fecha y hora de emisión del certificado; caso contrario se mostrará un mensaje rechazando la identidad de dicho ingresante, registrando la fecha y hora de rechazo.	

Tabla N° 4.25: Tareas de ingeniería. Emitir certificado de ingreso mediante verificación de identidad de ingresante.

<b>TAREAS DE INGENIERÍA</b>	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 7
<b>Nombre de tarea:</b> Registrar datos de docente	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 28/06/1016	<b>Fecha de fin:</b> 30/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar docente del menú Docente; el sistema muestra la interfaz de Registro de datos del docente, el usuario ingresa los datos del docente, luego hace clic en el botón Guardar, en caso los datos del docente se guarden con éxito, el sistema mostrará un mensaje de confirmación caso contrario mostrará un mensaje de error.	

Tabla N° 4.26: Tareas de ingeniería. Registrar datos de docente.

<b>TAREAS DE INGENIERÍA</b>	
<b>Número de tarea de ingeniería:</b> 2	<b>Número de historia de usuario:</b> 7
<b>Nombre de tarea:</b> Registrar huella de docente	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 28/06/1016	<b>Fecha de fin:</b> 30/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar docente del menú Docente; el sistema muestra la interfaz de Registro de datos del docente. El usuario puede hacer la	

búsqueda de los datos del docente ingresando el DNI correcto del docente encargado, en caso el docente no esté registrado, el usuario ingresa los datos de un docente, luego hace clic en el botón Guardar, y si el docente ya existe se puede realizar la modificación de datos en casos sea necesario. Para registrar la huella del docente en la misma ventana, se debe elegir el dedo del cual se registrará la huella, el docente colocará el dedo sobre la superficie del lector de huella digital, y el sistema capturará la huella, luego hace clic en el botón Registrar huella , si desea volver a tomar la huella debe volver a poner el dedo sobre el lector y se volverá a tomar la huella, luego hacer clic en Registrar huella, el sistema preguntará si se desea actualizar la huella, y finalmente para guardar los cambios se hace clic en el botón Guardar.

Tabla Nº 4.27: Tareas de ingeniería. Registrar huella de docente.

<b>TAREAS DE INGENIERÍA</b>	
<b>Número de tarea de ingeniería:</b> 3	<b>Número de historia de usuario:</b> 7
<b>Nombre de tarea:</b> Actualizar huella de docente	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 28/06/1016	<b>Fecha de fin:</b> 30/06/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Registrar docente del menú Docente; el sistema muestra la interfaz de Registro de datos del docente. El usuario puede hacer la búsqueda de los datos del docente ingresando el DNI correcto del docente encargado, en caso el docente no esté registrado, el usuario ingresa los datos de un docente, luego hace clic en el botón Guardar, y si el docente ya existe se puede realizar la modificación de datos en casos sea necesario. Para registrar la huella del docente en la misma ventana, se debe elegir el dedo del cual se registrará la huella, el docente colocará el dedo sobre la superficie del lector de huella digital, y el sistema capturará la huella, luego hace clic en el botón Registrar huella , si desea volver a tomar la huella debe volver a poner el dedo sobre el lector y se volverá a tomar la huella, luego hacer clic en Registrar huella, el sistema preguntará si se desea actualizar la huella, y finalmente para guardar los cambios se hace clic en el botón Guardar.	

Tabla Nº 4.28: Tareas de ingeniería. Actualizar huella de docente.

<b>TAREAS DE INGENIERÍA</b>	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 8
<b>Nombre de tarea:</b> Controlar asistencia de docente	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 01/07/1016	<b>Fecha de fin:</b> 05/07/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El docente, hace clic en la opción Asistencia del menú Docente; el sistema muestra la interfaz de Control de asistencia de docente, en la interfaz se muestra un marcador de hora en el que el docente podrá ingresar su DNI, luego el docente coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del docente, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del docente y registrando en la base de datos la fecha y hora de marcado de asistencia; caso contrario se mostrará un mensaje rechazando la identidad de dicho docente y el docente podrá repetir la acción.	

Tabla Nº 4.29: Tareas de ingeniería. Controlar asistencia de docente.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 9
<b>Nombre de tarea:</b> Emitir acta de notas mediante verificación de identidad de docente	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 4
<b>Fecha de inicio:</b> 06/07/1016	<b>Fecha de fin:</b> 11/07/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El docente, hace clic en la opción Acta de notas del menú Docente; el sistema muestra la interfaz de Verificación de docente, el docente ingresa su DNI la identidad, luego el docente coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del docente, en caso la identidad sea rechazada se mostrará un mensaje indicando que la verificación de identidad no se realizó con éxito, y si la identidad del docente es verificada se mostrará la ventana de Emisión de acta de notas, en el que el docente podrá seleccionar un curso que dicta y podrá realizar el registro de notas finales por alumnos matriculados en el curso seleccionado, una vez que el docente termine de registrar las notas finales, el docente podrá guardar el registro de notas haciendo clic en el botón Guardar, y en caso desee emitir el acta de notas deberá hacer clic en el botón Emitir acta, y el sistema le mostrará una ventana de confirmación en el que el docente deberá colocar sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realizará la verificación de la identidad del docente, en caso la identidad sea confirmada se emitirá el acta de notas del curso; caso contrario no se emitirá ningún acta de notas.	

Tabla Nº 4.30: Tareas de ingeniería. Emitir acta de notas mediante verificación de identidad de docente.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 10
<b>Nombre de tarea:</b> Verificar identidad de comensal	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 12/07/1016	<b>Fecha de fin:</b> 14/07/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Verificar comensal del menú Bienestar universitario; el sistema muestra la interfaz de Verificación de comensal, el usuario ingresa el DNI del estudiante a verificar la identidad, luego el estudiante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su huella, el sistema realiza la verificación de identidad del estudiante, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del estudiante y registrando en la base de datos la fecha y hora de acceso del estudiante al comedor; caso contrario se mostrará un mensaje rechazando la identidad de dicho estudiante, registrando la fecha y hora de rechazo.	

Tabla Nº 4.31: Tareas de ingeniería. Verificar identidad de comensal.

TAREAS DE INGENIERÍA	
<b>Número de tarea de ingeniería:</b> 1	<b>Número de historia de usuario:</b> 11
<b>Nombre de tarea:</b> Verificar identidad de usuario de biblioteca	
<b>Tipo de tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3
<b>Fecha de inicio:</b> 15/07/1016	<b>Fecha de fin:</b> 19/07/2016
<b>Programador responsable:</b> Juan Antonio Gabino Candia	
<b>Descripción:</b> El usuario, hace clic en la opción Verificar usuario de biblioteca del menú Biblioteca universitaria; el sistema muestra la interfaz de Verificación de usuario de biblioteca, el usuario ingresa el DNI del estudiante a verificar la identidad, luego el estudiante coloca sobre la superficie del lector de huella, uno de los dedos de los cuales se registró su	

huella, el sistema realiza la verificación de identidad del estudiante, en caso la identidad sea verificada se mostrará un mensaje de confirmación aceptando la identidad del estudiante y registrando en la base de datos la fecha y hora de solicitud de préstamo de material bibliográfico a domicilio; caso contrario se mostrará un mensaje rechazando la identidad de dicho estudiante, registrando la fecha y hora de rechazo.

Tabla N° 4.32: Tareas de ingeniería. Verificar identidad de usuario de biblioteca.

#### 4.1.4 PLAN DE ITERACIÓN

N°	HISTORIA DE USUARIO	TAREA DE INGENIERÍA
1	Registrar y mantener datos de postulante	Registrar datos del postulante
		Registrar huellas de postulante
		Actualizar datos de postulante
		Cargar fotografía de postulante
2	Mantener huella de postulante	Consultar registro de huella de postulante
		Eliminar registro de huella de postulante
3	Consultar datos de postulante	Buscar datos de postulante
4	Registrar y mantener proceso de admisión	Registrar proceso de admisión
		Definir el proceso de admisión
5	Verificar identidad de postulante	Verificar identidad de postulante mediante su huella al momento de acceso al Examen de Admisión
6	Emisión de certificado de ingreso	Emitir certificado de ingreso mediante verificación de identidad de ingresante
7	Registro y mantenimiento de datos del docente	Registrar datos de docente
		Registrar huella de docente
		Actualizar huella de docente
8	Controlar docente	Controlar asistencia de docente
9	Emisión de acta de notas	Emitir acta de notas mediante verificación de identidad de docente
10	Verificar identidad de comensal	Verificar identidad de comensal
11	Verificar identidad de usuario de biblioteca	Verificar identidad de usuario de biblioteca

Tabla N° 4.33: Plan de iteración (Primera).

HISTORIA DE USUARIO	TAREA DE ING.	FECHA DE INICIO	FECHA DE FIN	PROGRAMADOR
1	1	01/06/2016	08/06/2016	Juan Gabino Candia
	2	01/06/2016	08/06/2016	Juan Gabino Candia
	3	01/06/2016	08/06/2016	Juan Gabino Candia
	4	01/06/2016	08/06/2016	Juan Gabino Candia
2	1	09/06/2016	10/06/2016	Juan Gabino Candia
	2	09/06/2016	10/06/2016	Juan Gabino Candia
3	1	13/06/2016	14/06/2016	Juan Gabino Candia
4	1	15/06/2016	16/06/2016	Juan Gabino Candia
	2	15/06/2016	16/06/2016	Juan Gabino Candia
5	1	17/06/2016	22/06/2016	Juan Gabino Candia
6	1	23/06/2016	28/06/2016	Juan Gabino Candia
7	1	28/06/2016	30/09/2016	Juan Gabino Candia
	2	28/06/2016	30/09/2016	Juan Gabino Candia
	3	28/06/2016	30/09/2016	Juan Gabino Candia
8	1	01/07/2016	05/07/2016	Juan Gabino Candia
9	1	06/07/2016	11/07/2016	Juan Gabino Candia

10	1	12/07/2016	14/07/2016	Juan Gabino Candia
11	1	15/07/2016	19/07/2016	Juan Gabino Candia

Tabla N° 4.34: Plan de iteración, clasificados en fechas de desarrollo.

N° H.U.	REQUISITO	N° C.P.	CASO DE PRUEBA (C.P.)
1	El usuario es capaz de registrar los datos del postulante	1	El sistema validará que se ingresen los datos correctos del postulante.
	El usuario es capaz de registrar las huellas digitales de los postulantes	2	El sistema mostrará la imagen de la huella digital, y registrará la plantilla de la huella en la Base de Datos.
	El usuario es capaz de actualizar las huellas de postulantes registradas	3	El sistema permitirá al usuario actualizar las huellas previamente registradas de un postulante.
	El usuario, será capaz de actualizar datos del postulante.	4	El sistema permitirá actualizar los datos del postulante.
	El usuario es capaz de realizar la carga de la fotografía del postulante.	5	El sistema permitirá al usuario, realizar la carga de fotografía del postulante.
2	El usuario realiza la consulta de las huellas registradas del postulante.	1	El sistema mostrará las huellas registradas del postulante
	El usuario elimina las huellas registradas del postulante.	2	El sistema permitirá al usuario, eliminar las huellas registradas del postulante.
3	El usuario es capaz de consultar los datos del postulante.	1	El sistema permitirá al usuario, realizar la consulta de los datos del postulante.
4	El usuario es capaz de registrar un nuevo proceso de admisión.	1	El sistema permitirá al usuario, registrar un nuevo proceso de admisión.
	El usuario es capaz de elegir el proceso de admisión correcto.	2	El sistema permitirá al usuario seleccionar el proceso de admisión que corresponde.
5	El usuario es capaz de verificar la identidad del postulante.	1	El sistema permitirá al usuario, realizar la verificación de la identidad del postulante.
6	El usuario es capaz de emitir el certificado de ingreso del postulante.	1	El sistema permitirá al usuario emitir su certificado de ingreso, previa verificación de su identidad.
7	El usuario es capaz de registrar y mantener los datos de los docentes.	1	El sistema permitirá al usuario registrar y mantener los datos del docente.
	El usuario es capaz de registrar y mantener las huellas digitales de los docentes.	2	El sistema permitirá al usuario registrar y mantener las huellas digitales del docente.
8	Los docentes serán capaces de registrar su asistencia.	1	El sistema podrá registrar la asistencia de los docentes, mediante la verificación de su identidad.
9	El usuario emite el acta con las notas finales de los alumnos por cada curso dictado.	1	El sistema permitirá la emisión del registro de acta de notas, previa validación de identidad del docente.
10	El usuario es capaz de verificar la identidad de los usuarios de comedor	1	El sistema permitirá al usuario, realizar la verificación de la

	universitario.		identidad del usuario del comedor universitario.
11	El usuario es capaz de verificar la identidad de los usuarios de la biblioteca.	1	El sistema permitirá al usuario, realizar la verificación de la identidad del usuario de biblioteca.

Tabla N° 4.35: Casos de prueba de aceptación.

## 4.2. INTERFAZ DE USUARIO Y CODIFICACIÓN

Las interfaces se han diseñado considerando la usabilidad del sistema, que permiten tener interfaces amigables y, fáciles de usar.



Figura N° 4.4: Interfaz. Ventana de Login para el sistema de control biométrico.

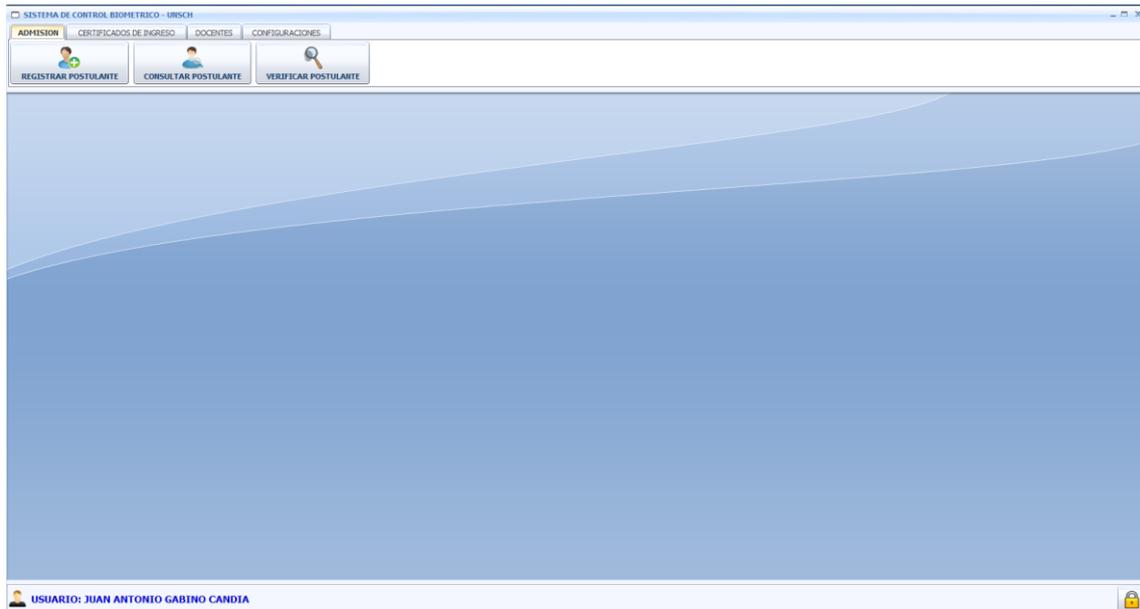


Figura N° 4.5: Interfaz. Pantalla inicial del sistema de control biométrico.

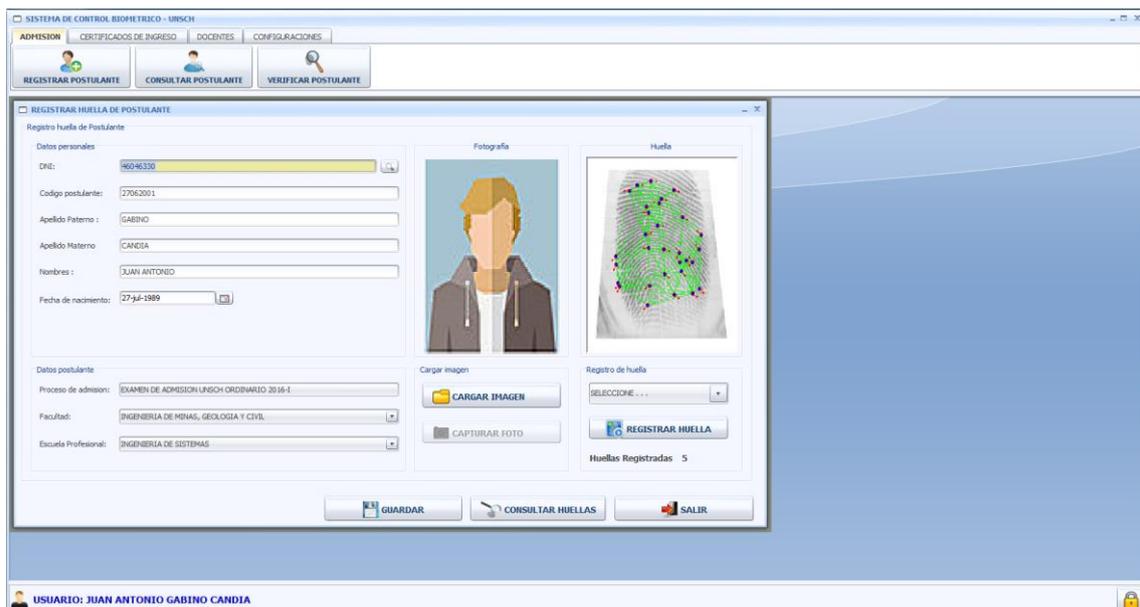


Figura N° 4.6: Interfaz. Ventana de registro de datos y huella del postulante.

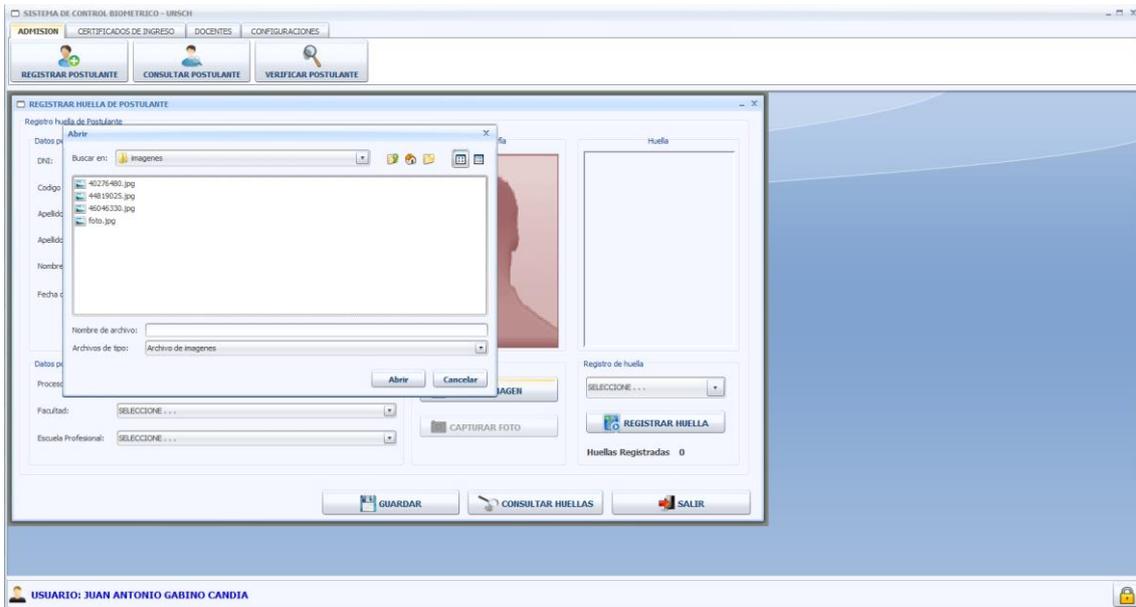


Figura N° 4.7: Interfaz. Ventana de carga de fotografía del postulante.

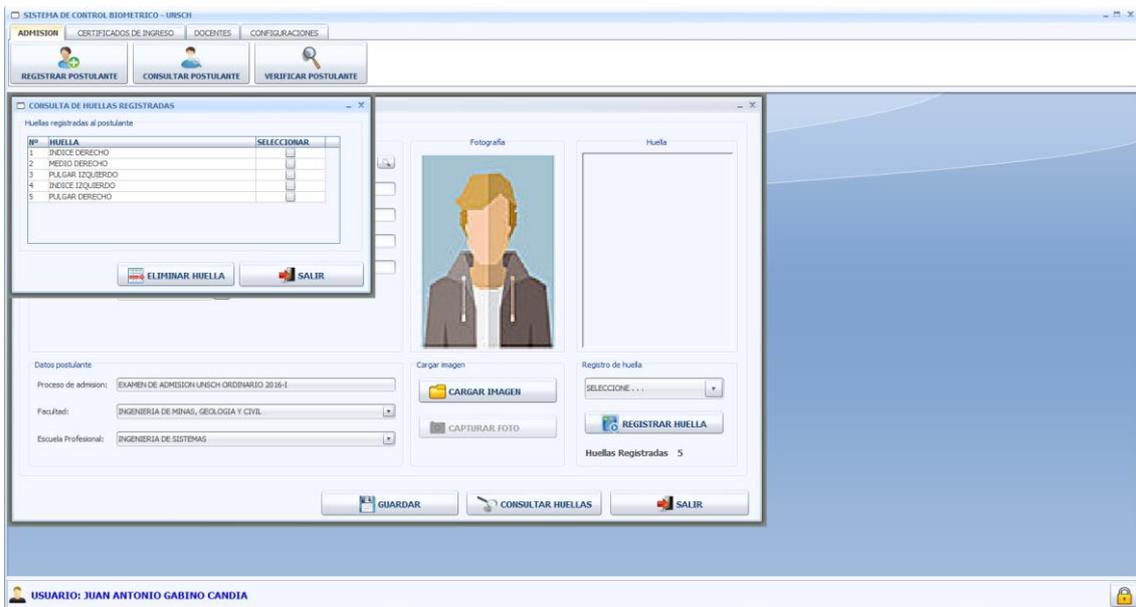


Figura N° 4.8: Interfaz. Ventana de consulta y mantenimiento de huella.



Figura N° 4.9: Interfaz. Ventana de verificación de identidad de postulante.

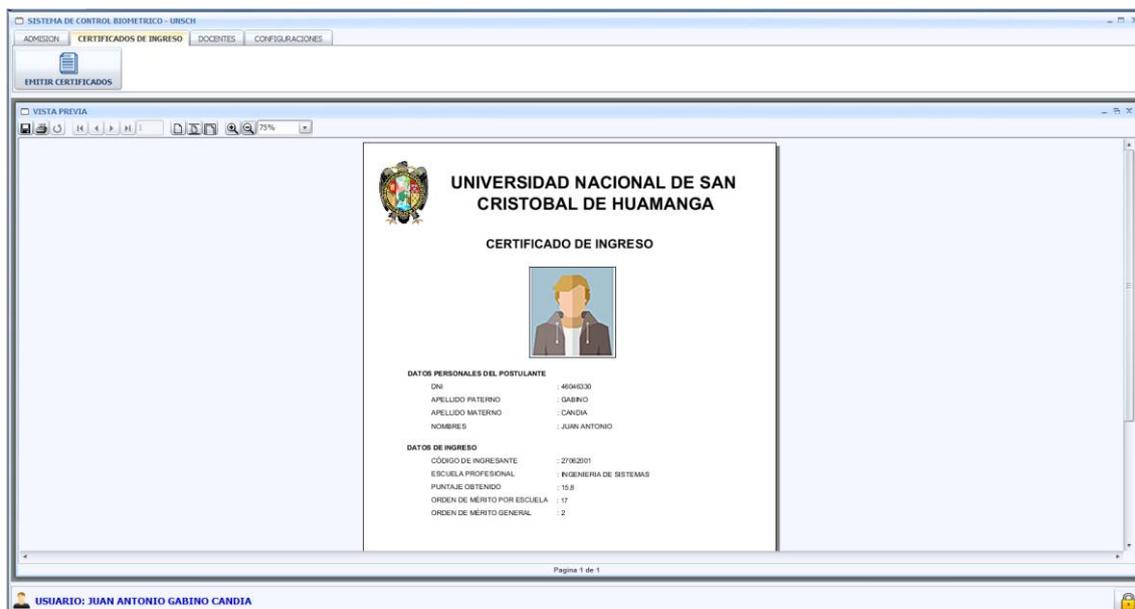


Figura N° 4.10: Interfaz. Ventana con el certificado de ingreso emitido.

### 4.3. TARJETAS CLASE RESPONSABILIDAD Y COLABORACIÓN (CRC)

Para un diseño simple, se procede a definir tarjetas CRC, que permitan identificar las clases, sus responsabilidades y las colaboraciones que realiza para una historia de usuario y las tareas correspondientes.

CLASE: AlumnoPostula	
<b>Responsabilidades:</b>	<b>Colaboradores:</b>
- Registrar datos del alumno.	- Postulante.

<ul style="list-style-type: none"> <li>- Registrar escuela a la que se inscribe le postulante.</li> <li>- Registrar huellas del postulante</li> <li>- Consultar datos del postulante.</li> </ul>	<ul style="list-style-type: none"> <li>- Escuela.</li> <li>- ProcesoAdmisión.</li> </ul>
--	--

Tabla N° 4.36: Clase AlumnoPostula.

<b>CLASE: AccesoExamenAdmision</b>	
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>- Verificar identidad del postulante.</li> <li>- Registrar acceso del postulante al examen de admisión.</li> <li>- Consultar accesos del postulante al examen.</li> </ul>	<b>Colaboradores:</b> <ul style="list-style-type: none"> <li>- Postulante.</li> </ul>

Tabla N° 4.37: Clase AccesoExamenAdmision.

<b>CLASE: CertificadoIngreso</b>	
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>- Verificar identidad del ingresante.</li> <li>- Emitir certificado de ingreso del postulante.</li> <li>- Consultar certificado de ingreso.</li> </ul>	<b>Colaboradores:</b> <ul style="list-style-type: none"> <li>- Postulante.</li> <li>- AlumnoPostula</li> </ul>

Tabla N° 4.38: Clase CertificadoIngreso.

<b>CLASE: Matricula</b>	
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>- Registrar matricula de alumno.</li> </ul>	<b>Colaboradores:</b> <ul style="list-style-type: none"> <li>- Alumno.</li> <li>- SemestreAcademico</li> <li>- Curso</li> </ul>

Tabla N° 4.39: Clase Matricula.

<b>CLASE: ActaNota</b>	
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>- Verificar identidad de docente.</li> <li>- Registrar notas finales de alumnos</li> <li>- Emitir actas de notas finales por cursos.</li> </ul>	<b>Colaboradores:</b> <ul style="list-style-type: none"> <li>- Curso.</li> <li>- Docente.</li> </ul>

Tabla N° 4.40: Clase ActaNota.

<b>CLASE: HorarioDocente</b>	
<b>Responsabilidades:</b> <ul style="list-style-type: none"> <li>- Registrar horario de docentes</li> <li>- Verificar asistencia de docentes al dictado de sus clases.</li> </ul>	<b>Colaboradores:</b> <ul style="list-style-type: none"> <li>- Horario.</li> <li>- Dia.</li> <li>- TipoClase.</li> <li>- Ambiente.</li> <li>- Docente.</li> </ul>

Tabla N° 4.41: Clase HorarioDocente.

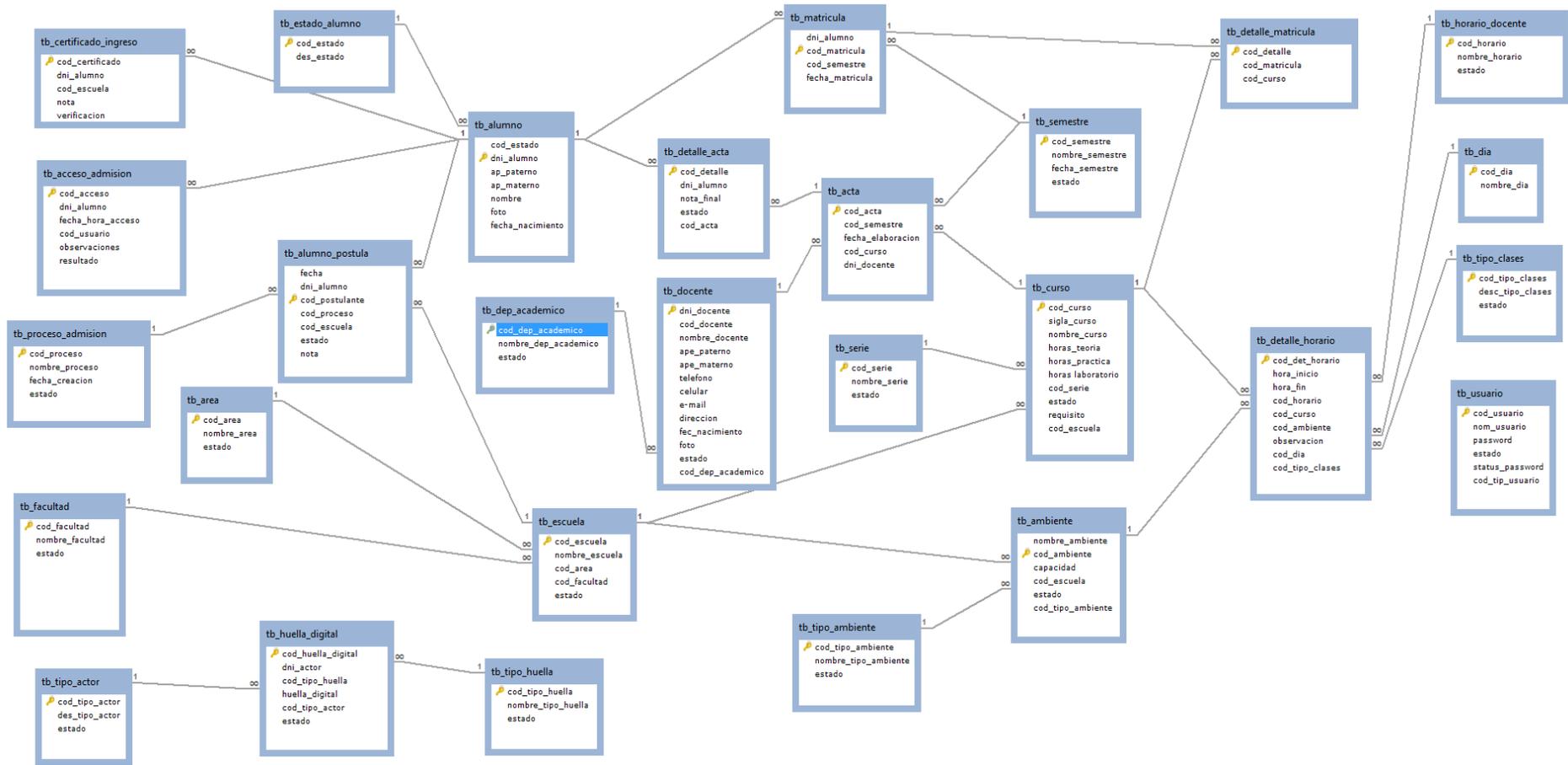


Figura N° 4.11: Modelo físico de base de datos



## 4.4. IMPLEMENTACIÓN

Implementación de la clase de lectura y verificación de la identidad de la huella.

```
package com.unsch.util;

import java.io.File;
import com.griaule.grfingerjava.FingerprintImage;
import com.griaule.grfingerjava.GrFingerJava;
import com.griaule.grfingerjava.GrFingerJavaException;
import com.griaule.grfingerjava.IFingerEventListener;
import com.griaule.grfingerjava.IImageEventListener;
import com.griaule.grfingerjava.IStatusEventListener;
import com.griaule.grfingerjava.MatchingContext;

public class LectorHuella implements IFingerEventListener, IImageEventListener, IStatusEventListener {

    /**Contexto utilizado para la captura, extracción y coincidencia de huellas digitales */
    protected MatchingContext sdk;

    public void onFingerDown(String idSensor) {
        //System.out.println("Dedo colocado");
    }

    public void onFingerUp(String idSensor) {
        //System.out.println("Dedo retirado");
    }

    public void onImageAcquired(String idSensor, FingerprintImage fi) {
    }

    //Conecta el lector de huella
    public void onSensorPlug(String idSensor) {
        System.out.println("Lector conectado");
        try {
            GrFingerJava.startCapture(idSensor, this, this);
        } catch (GrFingerJavaException e) {
            e.printStackTrace();
        }
    }

    //Desconecta el lector de huella
    public void onSensorUnplug(String idSensor) {
        try {
            GrFingerJava.stopCapture(idSensor);
        } catch (GrFingerJavaException e) {
            e.printStackTrace();
        }
    }

    //METODOS
    public static void setFingerprintSDKNativeDirectory(String directorio) {
        File directory = new File(directorio);

        try {
            GrFingerJava.setNativeLibrariesDirectory(directory);
            GrFingerJava.setLicenseDirectory(directory);
        } catch (GrFingerJavaException e) {
            e.printStackTrace();
        }
    }

    public void inicializarCaptura() {
        try {
            sdk = new MatchingContext();
            //Inicializa la captura de huella digital.
            GrFingerJava.initializeCapture(this);
        } catch (Exception e) {
            //Si ocurre un error se cierra la aplicación.
            e.printStackTrace();
            System.exit(1);
        }
    }

    public void finalizarCaptura() {
        try {
            sdk = new MatchingContext();
            //Finaliza la captura de huella digital.
            GrFingerJava.finalizeCapture();
        } catch (Exception e) {
            //Si ocurre un error se cierra la aplicación.
            e.printStackTrace();
            System.exit(1);
        }
    }
}
```

Figura N° 4.13: Código fuente. Clase que implementa la lectura de la huella.

```

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package com.unsch.util;

import com.griaule.grfingerjava.FingerprintImage;
import com.griaule.grfingerjava.GrFingerJava;
import com.griaule.grfingerjava.GrFingerJavaException;
import com.griaule.grfingerjava.Template;
import com.unsch.bean.AlumnoBean;
import com.unsch.bean.AlumnoPostulaBean;
import com.unsch.bean.HuellaDigitalBean;
import com.unsch.bean.TipoActorBean;
import com.unsch.controller.AlumnoController;
import com.unsch.controller.AlumnoPostulaController;
import com.unsch.controller.HuellaDigitalController;
import com.unsch.dao.AlumnoDao;
import com.unsch.dao.HuellaDigitalDao;
import com.unsch.view.VerificarHuellaPostulanteView;
import java.awt.Color;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

/**
 *
 * @author Administrador
 */
public class LectorVerificar extends LectorHuella {

    private VerificarHuellaPostulanteView ui;
    //La imagen de la última huella digital capturada
    private FingerprintImage imagenHuella;
    //La plantilla de la última imagen de huella capturada
    private Template plantillaHuellaActual;

    public LectorVerificar(VerificarHuellaPostulanteView ui) {
        this.ui = ui;
    }

    @Override
    public void onImageAcquired(String idSensor, FingerprintImage huellaDigital) {
        //Almacena la imagen de la huella
        imagenHuella = huellaDigital;
        //Muestra la imagen obtenida
        ui.showImage(huellaDigital);
        //Muestra la plantilla en la imagen actual
        extraerPlantilla();
    }

    //Evento cuando se retira el dedo del lector
    @Override
    //Evento cuando se retira el dedo del lector
    @Override
    public void onFingerUp(String idSensor) {
        if (VerificarHuellaPostulanteView.jtfDNI.getText().isEmpty()) {
            JOptionPane.showMessageDialog(null, "INGRESE EL DNI DEL POSTULANTE", "ALERTA", JOptionPane.WARNING_MESSAGE);
            VerificarHuellaPostulanteView.jtfDNI.requestFocus();
            return;
        }
        AlumnoBean alumnoBean = verificar(VerificarHuellaPostulanteView.jtfDNI.getText());

        if (alumnoBean != null) {
            try {
                VerificarHuellaPostulanteView.jtfNombres.setText(alumnoBean.getNombre());
                VerificarHuellaPostulanteView.jtfApellidos.setText(alumnoBean.getApPaterno() + " " + alumnoBean.getApMaterno());
                VerificarHuellaPostulanteView.jlMensaje.setText("POSTULANTE AUTORIZADO");
                VerificarHuellaPostulanteView.jlMensaje.setForeground(Color.BLUE);
                URL url = new URL("file://" + alumnoBean.getFoto());
                cargarFotoFileChooser(url, VerificarHuellaPostulanteView.jpFotografia);
                AlumnoPostulaController postulaController = new AlumnoPostulaController();
                AlumnoPostulaBean postulaBean = postulaController.consultarPostulante(VerificarHuellaPostulanteView.jtfDNI.getText());
                VerificarHuellaPostulanteView.jtfEscuela.setText(postulaBean.getEscuela().getNombreEscuela());
                VerificarHuellaPostulanteView.jtfCodPostulante.setText(postulaBean.getCodPostulante());
            } catch (MalformedURLException ex) {
                Logger.getLogger(LectorVerificar.class.getName()).log(Level.SEVERE, null, ex);
            }
        } else {
            VerificarHuellaPostulanteView.jtfNombres.setText("");
            VerificarHuellaPostulanteView.jtfApellidos.setText("");
            VerificarHuellaPostulanteView.jtfEscuela.setText("");
            VerificarHuellaPostulanteView.jtfCodPostulante.setText("");
            VerificarHuellaPostulanteView.jpFotografia = null;
            VerificarHuellaPostulanteView.jlMensaje.setText("POSTULANTE NO AUTORIZADO");
            VerificarHuellaPostulanteView.jlMensaje.setForeground(Color.RED);
        }
    }

    //Metodo para extraer la plantilla
    public void extraerPlantilla() {
        try {
            //Extrae la plantilla de la imagen.
            plantillaHuellaActual = sdk.extract(imagenHuella);
            //Muestra la plantilla en la imagen
            ui.showImage(GrFingerJava.getBimetricImage(plantillaHuellaActual, imagenHuella));
        } catch (GrFingerJavaException e) {
            e.printStackTrace();
        }
    }
}

```

```

public AlumnoBean verificar(String codigo) {
    AlumnoBean alumno = null;

    TipoActorBean actor = new TipoActorBean();
    actor.setCodTipoActor(AlumnoDao.POSTULANTE);

    HuellaDigitalBean huellaBean = new HuellaDigitalBean();
    huellaBean.setDni(codigo);
    huellaBean.setEstado(HuellaDigitalDao.ACTIVO);
    huellaBean.setTipoActor(actor);

    HuellaDigitalController huellaController = new HuellaDigitalController();
    ArrayList<HuellaDigitalBean> listHuellas = huellaController.consultarHuellas(huellaBean);

    ArrayList<Template> lPlantillas = new ArrayList<Template>();
    //Guardamos en un array temporal las plantillas de referencia almacenadas en la Base de datos
    for (HuellaDigitalBean huellaDigital: listHuellas) {
        Template template = new Template(huellaDigital.getHuellaDigital().getPlantilla());
        lPlantillas.add(template);
    }

    //Comparamos la plantilla actual obtenida con las de la Base de datos
    try {
        if (!lPlantillas.isEmpty()) {
            for (Template templateReference : lPlantillas) {
                if (templateReference != null && sdk.verify(plantillaHuellaActual, templateReference)) {
                    AlumnoController controller = new AlumnoController();
                    alumno = controller.consultarDatosAlumno(codigo);
                    break;
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return alumno;
}

private void cargarFotoFileChooser(URL url, JPanel jPanel) {
    PanelImagen panelImagen = new PanelImagen();
    panelImagen.cargarFotoUrl(url, jPanel);
}
}

```

Figura N° 4.14: Código fuente. Clase que implementa la verificación de la huella.

#### 4.5. CASOS DE PRUEBA Y ACEPTACIÓN

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 01	<b>Tarea de ingeniería:</b> Registrar datos del postulante
<b>Descripción:</b> El sistema validará que se ingresen los datos correctos del postulante.	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado en el sistema.	
<b>Entrada / Pasos de ejecución:</b> El usuario ingresa los datos del postulante. El sistema valida los campos obligatorios.	
<b>Resultado esperado:</b> El sistema guarda los datos del postulante y muestra un mensaje de confirmación.	
<b>Evaluación de la prueba:</b> Se debe mostrar un mensaje de confirmación.	

Tabla N° 4.42: Caso de prueba. Registrar datos del postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 02	<b>Tarea de ingeniería:</b> Registrar huellas de postulante
<b>Descripción:</b> El usuario es capaz de registrar las huellas digitales de los postulantes	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado en el sistema. El lector de huella debe estar conectado a la computadora.	
<b>Entrada / Pasos de ejecución:</b> Ingresar el DNI del postulante.	

Elegir el dedo a registrar la huella. Colocar sobre la superficie del lector de huella el dedo seleccionado Clic en el botón de agregar huella.
<b>Resultado esperado:</b> El sistema mostrará en la ventana la imagen de la huella
<b>Evaluación de la prueba:</b> Se debe mostrar un mensaje de confirmación de registro

Tabla Nº 4.43: Caso de prueba. Registrar huellas de postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 03	<b>Tarea de ingeniería:</b> Actualizar datos de postulante
<b>Descripción:</b> El usuario es capaz de actualizar los datos de un postulante ya registrado.	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado en el sistema. Deben existir datos del postulante.	
<b>Entrada / Pasos de ejecución:</b> Buscar un postulante a partir de su DNI.	
<b>Resultado esperado:</b> El sistema guarda los datos actualizados del postulante y muestra un mensaje de confirmación.	
<b>Evaluación de la prueba:</b> Se debe mostrar un mensaje de confirmación.	

Tabla Nº 4.44: Caso de prueba. Actualizar datos de postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 04	<b>Tarea de ingeniería:</b> Cargar fotografía de postulante
<b>Descripción:</b> El usuario es capaz de cargar la fotografía del postulante.	
<b>Condiciones de ejecución:</b> Tener registrado un postulante Tener una imagen del postulante.	
<b>Entrada / Pasos de ejecución:</b> El usuario busca un postulante a partir de su DNI. El sistema muestra los datos del postulante. El usuario hace clic en el botón de cargar imagen.	
<b>Resultado esperado:</b> El usuario debe poder seleccionar la imagen de la fotografía que le corresponde al postulante	
<b>Evaluación de la prueba:</b> El sistema muestra la fotografía del postulante.	

Tabla Nº 4.45: Caso de prueba. Cargar fotografía de postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 05	<b>Tarea de ingeniería:</b> Consultar registro de huella de postulante
<b>Descripción:</b> El usuario es capaz de consultar las huellas registradas del postulante.	
<b>Condiciones de ejecución:</b> Debe tener registrado datos de postulantes con sus respectivas huellas.	
<b>Entrada / Pasos de ejecución:</b> El usuario debe buscar un los datos de un postulante a partir de su DNI. El usuario debe hacer clic en el botón consultar huellas.	
<b>Resultado esperado:</b>	

El sistema muestra una ventana que contiene una grilla con los datos de las huellas registradas del postulante.
<b>Evaluación de la prueba:</b> Se muestra una grilla con las huellas registradas del postulante.

Tabla N° 4.46: Caso de prueba. Consultar registro de huella de postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 06	<b>Tarea de ingeniería:</b> Eliminar registro de huella de postulante
<b>Descripción:</b> El usuario será capaz de eliminar huellas registradas del postulante.	
<b>Condiciones de ejecución:</b> El usuario debe buscar los datos de un postulante a partir de su DNI. El usuario debe hacer clic en el botón de consultar huellas El sistema muestra una ventana con la grilla de huellas registradas.	
<b>Entrada / Pasos de ejecución:</b> El usuario selecciona una huella consultada. El usuario hace clic en el botón de eliminar huella.	
<b>Resultado esperado:</b> El sistema elimina la huella registrada del postulante.	
<b>Evaluación de la prueba:</b> Ya no se visualizará en la grilla la huella eliminada.	

Tabla N° 4.47: Caso de prueba. Eliminar registro de huella de postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 07	<b>Tarea de ingeniería:</b> Buscar datos de postulante
<b>Descripción:</b> El usuario será capaz de buscar los datos de un postulante.	
<b>Condiciones de ejecución:</b> Tener registrado datos de postulantes	
<b>Entrada / Pasos de ejecución:</b> El usuario realiza la búsqueda de los datos de un postulante a partir de su DNI.	
<b>Resultado esperado:</b> El sistema muestra los datos del postulante buscado	
<b>Evaluación de la prueba:</b> Se muestran los datos del postulante buscado	

Tabla N° 4.48: Caso de prueba. Buscar datos de postulante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 08	<b>Tarea de ingeniería:</b> Registrar proceso de admisión
<b>Descripción:</b> El usuario será capaz registrar el proceso de admisión actual.	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado.	
<b>Entrada / Pasos de ejecución:</b> El usuario debe seleccionar la opción sistema del menú configuración. En la ventana de registro de proceso de admisión, debe seleccionar la opción de agregar. El usuario debe ingresar los datos del proceso de admisión.	
<b>Resultado esperado:</b> El sistema muestra un mensaje de confirmación.	
<b>Evaluación de la prueba:</b> El sistema muestra en el combo, el nuevo proceso registrado.	

Tabla N° 4.49: Caso de prueba. Registrar proceso de admisión

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 09	<b>Tarea de ingeniería:</b> Definir el proceso de admisión
<b>Descripción:</b> El usuario será capaz seleccionar el proceso de admisión actual.	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado.	
<b>Entrada / Pasos de ejecución:</b> El usuario debe seleccionar la opción sistema del menú configuración. El sistema muestra la ventana de registro de proceso de admisión. El usuario debe seleccionar el proceso de admisión. El usuario debe hacer clic en guardar.	
<b>Resultado esperado:</b> El sistema muestra un mensaje de confirmación.	
<b>Evaluación de la prueba:</b> El sistema se cierra y se configura con el proceso de admisión definido.	

Tabla Nº 4.50: Caso de prueba. Definir el proceso de admisión

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 10	<b>Tarea de ingeniería:</b> Verificar identidad de postulante mediante su huella al momento de acceso al Examen de Admisión
<b>Descripción:</b> El usuario podrá verificar la identidad de los postulantes al momento de acceder al examen de admisión.	
<b>Condiciones de ejecución:</b> El usuario debe seleccionar la opción verificar postulante. El usuario debe ingresar el DNI del postulante.	
<b>Entrada / Pasos de ejecución:</b> El postulante coloca el dedo sobre el lector de huella.	
<b>Resultado esperado:</b> El sistema muestra el resultado de la verificación de postulante, y registra la fecha y hora de acceso al examen de admisión.	
<b>Evaluación de la prueba:</b> En caso la verificación sea exitosa se muestran los datos del postulante.	

Tabla Nº 4.51: Caso de prueba. Verificar identidad de postulante mediante su huella al momento de acceso al Examen de Admisión

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 11	<b>Tarea de ingeniería:</b> Emitir certificado de ingreso mediante verificación de identidad de ingresante
<b>Descripción:</b> El usuario será capaz de emitir el certificado de ingreso de los postulantes que ingresaron.	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado. Tener postulantes ingresantes.	
<b>Entrada / Pasos de ejecución:</b> Ingresar a la opción certificado de ingreso El usuario debe ingresar el DNI del postulante ingresante. El postulante ingresante debe colocara su dedo sobre el lector de huella. El sistema verifica la identidad del postulante ingresante y habilita la opción emitir certificado	
<b>Resultado esperado:</b> El sistema muestra el certificado con la opción de impresión.	
<b>Evaluación de la prueba:</b> Se muestra el certificado de ingreso	

Tabla Nº 4.52: Caso de prueba. Emitir certificado de ingreso mediante verificación de identidad de ingresante

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 12	<b>Tarea de ingeniería:</b> Registrar datos de docente
<b>Descripción:</b> El sistema validará que se ingresen los datos correctos del docente.	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado en el sistema.	
<b>Entrada / Pasos de ejecución:</b> El usuario ingresa los datos del docente. El sistema valida los campos obligatorios.	
<b>Resultado esperado:</b> El sistema guarda los datos del docente y muestra un mensaje de confirmación.	
<b>Evaluación de la prueba:</b> Se debe mostrar un mensaje de confirmación.	

Tabla N° 4.53: Caso de prueba. Registrar datos de docente

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 13	<b>Tarea de ingeniería:</b> Registrar huella de docente
<b>Descripción:</b> El usuario es capaz de registrar las huellas digitales de los docentes	
<b>Condiciones de ejecución:</b> El usuario debe estar logueado en el sistema. El lector de huella debe estar conectado a la computadora.	
<b>Entrada / Pasos de ejecución:</b> Ingresar el DNI del docente. Elegir el dedo a registrar la huella. Colocar sobre la superficie del lector de huella el dedo seleccionado Clic en el botón de agregar huella.	
<b>Resultado esperado:</b> El sistema mostrará en la ventana la imagen de la huella.	
<b>Evaluación de la prueba:</b> Se debe mostrar un mensaje de confirmación de registro.	

Tabla N° 4.54: Caso de prueba. Registrar huella de docente

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 14	<b>Tarea de ingeniería:</b> Actualizar huella de docente
<b>Descripción:</b> El usuario será capaz de actualizar las huellas registradas del docente.	
<b>Condiciones de ejecución:</b> El usuario debe buscar los datos de un docente a partir de su DNI. El usuario selecciona una huella del docente previamente registrado. El docente coloca su dedo sobre la superficie de lector de huella.	
<b>Entrada / Pasos de ejecución:</b> El usuario hace clic en el botón agregar huella. El sistema muestra un mensaje de confirmación si se desea actualizar la huella.	
<b>Resultado esperado:</b> El sistema actualiza la huella registrada del docente.	
<b>Evaluación de la prueba:</b> Se muestra un mensaje de confirmación de actualización de huella.	

Tabla N° 4.55: Caso de prueba. Actualizar huella de docente

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 15	<b>Tarea de ingeniería:</b> Controlar asistencia de docente

<b>Descripción:</b> El sistema será capaz de registrar la asistencia del docente a sus diversas actividades.
<b>Condiciones de ejecución:</b> Los datos del docente, incluidos sus huellas deben estar previamente registrados en el sistema. Debe estar abierta la ventana de control de asistencia
<b>Entrada / Pasos de ejecución:</b> El docente digita su número de DNI y coloca su dedo sobre la superficie del lector de huella
<b>Resultado esperado:</b> El sistema verifica la identidad del docente, y registra la asistencia del docente.
<b>Evaluación de la prueba:</b> Se muestra un mensaje de confirmación de registro de asistencia.

Tabla N° 4.56: Caso de prueba. Controlar asistencia de docente

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 16	<b>Tarea de ingeniería:</b> Emitir acta de notas mediante verificación de identidad de docente
<b>Descripción:</b> El usuario docente universitario, será capaz de emitir el acta de notas, previamente verificando su identidad.	
<b>Condiciones de ejecución:</b> Los datos del docente, incluidos los datos de sus huellas deben estar registrados en el sistema Debe existir un registro de calificaciones.	
<b>Entrada / Pasos de ejecución:</b> El usuario docente selecciona la opción de registro de notas. El docente coloca su dedo sobre la superficie del lector de huella. Hace clic en verificar identidad. En caso el sistema verifica la identidad, se habilita el botón de emitir acta de notas.	
<b>Resultado esperado:</b> El sistema emite el acta de notas, con la opción de imprimir.	
<b>Evaluación de la prueba:</b> Se visualiza el acta de notas.	

Tabla N° 4.57: Caso de prueba. Emitir acta de notas mediante verificación de identidad de docente

CASO DE PRUEBA DE ACEPTACIÓN	
<b>Código:</b> 17	<b>Tarea de ingeniería:</b> Verificar identidad de comensal
<b>Descripción:</b> El usuario podrá verificar la identidad de los estudiantes comensales al momento de hacer uso de los servicios del comedor universitario.	
<b>Condiciones de ejecución:</b> El usuario debe seleccionar la opción verificar identidad de comensal. El usuario debe ingresar el DNI del comensal.	
<b>Entrada / Pasos de ejecución:</b> El usuario ingresa el DNI del comensal. El comensal coloca el dedo sobre el lector de huella.	
<b>Resultado esperado:</b> El sistema muestra el resultado de la verificación del comensal, y registra la fecha y hora de acceso al comedor.	
<b>Evaluación de la prueba:</b> En caso la verificación sea exitosa se muestran los datos del comensal.	

Tabla N° 4.58: Caso de prueba. Verificar identidad de comensal

<b>CASO DE PRUEBA DE ACEPTACIÓN</b>	
<b>Código:</b> 18	<b>Tarea de ingeniería:</b> Verificar identidad de usuario de biblioteca
<b>Descripción:</b> El usuario podrá verificar la identidad de los estudiantes usuarios de biblioteca al momento de hacer uso del servicio de biblioteca.	
<b>Condiciones de ejecución:</b> El usuario debe seleccionar la opción verificar identidad de usuario de biblioteca. El usuario debe ingresar el DNI del usuario de biblioteca.	
<b>Entrada / Pasos de ejecución:</b> El usuario ingresa el DNI del usuario de biblioteca. El usuario de biblioteca coloca el dedo sobre el lector de huella.	
<b>Resultado esperado:</b> El sistema muestra el resultado de la verificación del usuario de biblioteca, y registra la fecha y hora de préstamo de material bibliográfico.	
<b>Evaluación de la prueba:</b> En caso la verificación sea exitosa se muestran los datos del usuario de biblioteca.	

Tabla N° 4.59: Caso de prueba. Verificar identidad de usuario de biblioteca

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1. CONCLUSIONES**

- a) Según los artefactos desarrollados en el capítulo IV, las historias de usuario definidas en la Tabla N° 4.1 de la sección 4.1.1, las historias de usuarios desarrolladas en las Tablas N° 4.12 y 4.13, el modelo físico de base de datos en la figura N° 4.11 y el diagrama de clases en la figura N° 4.12, se concluye que se logró el objetivo de verificar la identidad de los estudiantes, usuarios de biblioteca, comensales y usuarios del autoseguro, al ejecutar los procesos críticos de gestión universitaria.
- b) Según los artefactos desarrollados en el capítulo IV, las historias de usuario definidas en la Tabla N° 4.1 de la sección 4.1.1, las historias de usuarios desarrolladas en las Tablas N° 4.9, 4.10 y 4.11, el modelo físico de base de datos en la figura N° 4.11 y el diagrama de clases en la figura N° 4.12, se concluye que se lograron los objetivos de verificar la identidad de los docentes universitarios y garantizar que los documentos de gestión, sean emitidos por los docentes que corresponden.
- c) Según los artefactos desarrollados en el capítulo IV, las historias de usuario definidas en la Tabla N° 4.1 de la sección 4.1.1, las historias de usuarios desarrolladas en las Tablas N° 4.3, 4.4, 4.5, 4.7 y 4.8, el modelo físico de base de datos en la figura N° 4.11 y el diagrama de clases en la figura N° 4.12, se concluye que se logró el objetivo de verificar la identidad de los postulantes, al ejecutar los procesos críticos de gestión universitaria.

## **5.2. RECOMENDACIONES**

- a) Se debe investigar otras técnicas biométricas, tales como reconocimiento de rostro, geometría de la mano, reconocimiento de iris, patrones de retina o reconocimiento de voz, para realizar la verificación de identidad de los actores que ejecutan los procesos críticos de gestión universitaria en la Universidad Nacional de San Cristóbal de Huamanga.
- b) Investigar el desarrollo un algoritmo para el reconocimiento de huellas dactilares, con el fin de buscar el 0 % de error al realizar la identificación de personas.
- c) Investigar la forma de desarrollar el aplicativo de identificación de actores de la universidad, pero para dispositivos móviles con lectores de huellas digitales integrados, con el fin de garantizar la identidad de los actores que ejecutan los procesos críticos de gestión universitaria.

## BIBLIOGRAFÍA

1. Aguilar, A. (2002). *Introducción a la programación extrema*. Recuperado el 25 de noviembre de 2011, de <http://www.willydev.net/descargas/articulos/general/IntroXP.PDF>
2. Balmelli, L. (2006). *Verificación de Identidad de Personas mediante Sistemas Biométricos para el Control de Acceso a una Universidad*. Tesis, Pontificia Universidad Católica del Perú, Lima, Perú.
3. Bernal, C. (2006). *Metodología de la investigación. Para administración, economía, humanidades y ciencias sociales* (2ª Ed.). México, PEARSON EDUCACIÓN.
4. Berrueta, D. (2006). *Programación extrema y software libre*. Recuperado el 25 de noviembre de 2011, de [http://www-lsi.die.upm.es/~carreras/ISSE/programacion\\_extrema\\_2.x2.pdf](http://www-lsi.die.upm.es/~carreras/ISSE/programacion_extrema_2.x2.pdf)
5. Carrión, R. (2009). *Desarrollo de un algoritmo de clasificación de la huella dactilar para la policía nacional del Perú*. Tesis, Pontificia Universidad Católica del Perú, Lima, Perú.
6. Estatuto, Adecuado a la ley universitaria N° 30220, 2015.
7. Flores, R. (2005), *Algoritmos, estructuras de datos y programación orientada a objetos* (3ª Ed.). México, ESIC Editorial.
8. Galvis, C. (2007). *Introducción a la biometría*. Recuperado el 19 de noviembre de 2011 de <http://www.monografias.com/trabajos43/biometria/biometria.shtml>
9. Gomez, A. (2007). *Enciclopedia de la Seguridad Informática*. (1º Ed.). México, Alfaomega.
10. González, A. (s.f). *Reconocimiento Automático De Pisadas*. Recuperado el 25 de noviembre de 2011, de <http://bibing.us.es/proyectos/abreproy/11405/fichero/RECONOCIMIENTO+AUTOMATICO+DE+PISADAS.pdf>
11. Infinite Consulting C.A. (s.f.). *Glosario*. Recuperado el 22 de noviembre de 2011, de <http://www.infiniteconsulting.com.ve/glosario.htm>

12. James, A. y Gerge M. (2003). *Sistema de Información Gerencial*. (7ª Ed). México, McGraw Hill Interamericana.
13. Joyanes L. (2002) *Programación en Java 2*. España, Ediciones McGraw - Hill.
14. Lopez, J. (s.f.). *Algoritmo para la Identificación de personas basadas en la huella dactilar*. Departamento de Ingeniería Electrónica.
15. Manual de Organización y Funciones, Oficina de Racionalización, Universidad Nacional de San Cristóbal de Huamanga, 2011.
16. NEOTEC. (s.f.). *Introducción a los Biométricos*. Recuperado el 24 de noviembre de 2011, de <http://www.neotec.com.pa/pdf/introduccionalosbiometricos.pdf>
17. NEOTEC (s.f.). *Los biométricos y la exactitud de la planilla*. (s.f.). Recuperado el 20 de noviembre de 2011, de <http://www.neotec.com.pa/pdf/bioplanilla.pdf>
18. Osorio, L. (2004). *Lógica y programación orientada a los objetos*. (3ª Ed.). España, SRM Editorial.
19. Reglamento de Organización y Funciones, PROPUESTA ELABORADA POR LAS COMISIONES DE REESTRUCTURACIÓN ACADÉMICA Y ADMINISTRATIVA, Universidad Nacional de San Cristóbal de Huamanga, 2014.
20. Simón, D. (2003). *Reconocimiento mediante patrones biométricos de huella dactilar*. Tesis Doctoral. Universidad Politécnica de Madrid, Madrid, España.
21. Terán, A. (2007). *Concepto de Base de datos Relacionales*. Recuperado el 22 de noviembre de 2011, de <http://basededatosrelacionales.blogspot.com/2007/10/concepto-de-base-de-datos-relacionales.html>
22. Texto Único de Procedimiento Administrativo, Oficina de Racionalización, Universidad Nacional de San Cristóbal de Huamanga, 2008
23. Tolosa, C. y Giz, A. (s.f.). *Sistemas Biométricos*. Recuperado el 17 de noviembre de 2011, de [www.dsi.uclm.es/asignaturas/Biometria/Trabajo%20Biometria.pdf](http://www.dsi.uclm.es/asignaturas/Biometria/Trabajo%20Biometria.pdf)
24. Wikipedia. (2011). *Base de datos relacional*. Recuperado el día 21 de noviembre de 2011, de <http://es.wikipedia.org/wiki/>

Base\_de\_datos\_relacional.

25. Wikipedia. (2011). *Biometría*. Recuperado el día 20 de noviembre de 2011, de <http://es.wikipedia.org/wiki/Biometría>

## ANEXOS

### ANEXO A: MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES

Variables	Dimensiones	Indicadores	Items	Instrumento
V1: Procesos críticos de gestión universitaria	1.1 Gestión académica y administrativa	Ley universitaria	¿Cuáles son los procesos críticos identificados en la ley universitaria?	Ficha de análisis documental
		Currículo de estudios	¿Cuáles son los procesos críticos identificados en el currículo de estudios de cada escuela de formación profesional?	Ficha de análisis documental
		Texto único de procedimiento administrativo	¿Cuáles son los procesos críticos identificados en el Texto Único de Procedimiento Administrativo?	Ficha de análisis documental
		Manual de organización y función	¿Cuáles son los procesos críticos identificados en el Manual de Organización y Funciones?	Ficha de análisis documental
		Reglamento de organización y funciones	¿Cuáles son los procesos críticos identificados en el Reglamento de Organización y Funciones?	Ficha de análisis documental
		Estatuto universitario	¿Cuáles son los procesos críticos identificados en el estatuto universitario?	Ficha de análisis documental
		Prospecto de admisión	¿Cuáles son los procesos críticos identificados en el prospecto de admisión?	Ficha de análisis documental

V2: Actor	2.1 Postulante	Suplantaciones	¿Cuáles son las técnicas de suplantación usadas?	Guía para entrevista
		Acceso al examen de admisión	¿Cuáles son las medidas actuales que se toma para el control de acceso al examen de admisión?	Guía para entrevista
			¿Cuál es el tiempo promedio que demora la identificación de un postulante?	Guía para entrevista
	2.2 Docente universitario	Asistencia a actividades del docente	¿Cómo se controla la asistencia del docente, al dictado de sus clases?	Cuestionario
			¿Cómo se controla la asistencia del docente para la atención a estudiantes?	Cuestionario
			¿Cómo se controla la asistencia del docente a labores de proyección social?	Cuestionario
			¿Cómo se controla la asistencia del docente a sesiones diversas?	Cuestionario
			¿Cómo se controla la asistencia del docente a comisiones diversas?	Cuestionario
			¿Cómo se controla la asistencia del docente a labores de investigación?	Cuestionario
		Documentos de emitidos	¿Cuáles son los documentos que emite un docente?	Cuestionario
			¿Qué documentos requieren que se valide, que el emisor es el docente correcto?	Cuestionario
	2.3 Estudiante	Bienestar universitario	¿Cuál es la concurrencia de los estudiantes al seguro universitario?	Cuestionario
			¿Cuál es la concurrencia de los estudiantes al comedor universitario?	Cuestionario
		Bibliotecas	¿Cuál es la concurrencia de los estudiantes a la biblioteca universitaria?	Cuestionario
		Identidad de nuevos ingresantes	¿Cuáles son las medidas actuales que se toma para la identificación de los nuevos ingresantes a la Universidad?	Guía para entrevista
¿Cuál es el tiempo promedio que demora la identificación de un ingresante?			Guía para entrevista	

Tabla N° A: Matriz de operacionalización de variables

## ANEXO B: FICHA DE ANÁLISIS DOCUMENTAL

<b>FICHA PARA ANÁLISIS DOCUMENTAL</b>	
Nombre de documento	
Fecha	
Procesos críticos identificados	
Descripción del proceso	Automatizable
<b>Total de procesos automatizables</b>	

Tabla N° B: Plantilla para ficha para análisis documental

## ANEXO C: GUÍA PARA ENTREVISTA

<p>1. ¿Cuáles son las técnicas de suplantación usadas?</p> <p>.....</p> <p>.....</p> <p>.....</p>
<p>2. ¿Cuáles son las medidas actuales que se toma para el control de acceso al examen de admisión?</p> <p>.....</p> <p>.....</p> <p>.....</p>
<p>3. ¿Cuál es el tiempo promedio que demora la identificación de un postulante?</p> <p>.....</p> <p>.....</p> <p>.....</p>
<p>4. ¿Cuáles son las medidas actuales que se toma para la identificación de los nuevos ingresantes a la Universidad?</p> <p>.....</p> <p>.....</p> <p>.....</p>
<p>5. ¿Cuál es el tiempo promedio que demora la identificación de un ingresante?</p> <p>.....</p> <p>.....</p> <p>.....</p>

Tabla N° C: Guía de entrevista para el presidente de la comisión del examen de admisión.

## ANEXO D: CUESTIONARIO

1. ¿Cómo se controla la asistencia del docente, al dictado de sus clases? ..... .....
2. ¿Cómo se controla la asistencia del docente para la atención a estudiantes? ..... .....
3. ¿Cómo se controla la asistencia del docente a labores de proyección social? ..... .....
4. ¿Cómo se controla la asistencia del docente a sesiones diversas? ..... .....
5. ¿Cómo se controla la asistencia del docente a comisiones diversas? ..... .....
6. ¿Cómo se controla la asistencia del docente a labores de investigación? ..... .....
7. ¿Cuáles son los documentos que emite un docente? ..... .....
8. ¿Qué documentos requieren que se valide, que el emisor es el docente correcto? ..... .....

Tabla N° D.1: Cuestionario para los docentes universitarios

1. ¿Cuál es la concurrencia de los estudiantes al seguro universitario? ..... .....
2. ¿Cuál es la concurrencia de los estudiantes al comedor universitario? ..... .....
3. ¿Cuál es la concurrencia de los estudiantes a la biblioteca universitaria? ..... .....

Tabla N° D.2: Cuestionario para el personal de biblioteca y bienestar universitario.