

**UNIVERSIDAD NACIONAL DEL ALTIPLANO - PUNO**

**FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**

**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**



**“SOFTWARE DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO  
EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO - 2008”**

**TESIS**

**PRESENTADA POR**

**Bach. ROXANA GALARZA LERMA**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

**PUNO - PERÚ**

**2009**

**UNIVERSIDAD NACIONAL DEL ALTIPLANO PUNO**  
**FACULTAD DE INGENIERIA ESTADÍSTICA E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERIA ESTADÍSTICA E INFORMÁTICA**



**“SOFTWARE DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI - UNA - PUNO – 2008”**

**TESIS**

Presentada por:

**Bach. Roxana Galarza Lerma**

Para optar el Título Profesional de:

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

APROBADA POR:

PRESIDENTE

:   
M. C. CONFESOR MILAN VARGAS VALVERDE

PRIMER MIEMBRO

: \_\_\_\_\_  
M.Sc. GODOFREDO QUISPE MAMANI

SEGUNDO MIEMBRO

:   
M.Sc. PEDRO LEONARDO QUISPE TICONA

DIRECTOR

:   
M.Sc. ALEJANDRO APAZA TARQUI

ÁREA

: Informática

TEMA

: Ingeniería del software

DEDICATORIA

*A Dios, a mis padres Adrián y Francisca, a  
Walter, por la ayuda y amor que me  
brindaron.*

**AGRADECIMIENTOS**

*Expreso mi más sincera gratitud a todas las personas quienes contribuyeron al desarrollo de este trabajo, brindándome valiosas sugerencias, apoyo moral y material.*

*A Dios por la salud que me brinda, a mis padres Adrián y Francisca y a Walter, quienes me apoyaron me motivaron en todo el tiempo de mis estudios y de mi tesis.*

*A mis docentes, que me apoyaron, me brindaron su tiempo y sus conocimientos para ayudarme a concluir satisfactoriamente mis estudios profesionales.*

**INDICE GENERAL**

RESUMEN .....	10
ABSTRACT .....	11
INTRODUCCIÓN .....	12

**CAPÍTULO I****PLAN DE INVESTIGACIÓN**

1.1. PLANTEAMIENTO Y DEFINICIÓN DEL PROBLEMA .....	13
1.2. OBJETIVO GENERAL Y ESPECÍFICOS .....	15
1.3. HIPÓTESIS .....	16
1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN .....	16
1.5. DELIMITACIÓN DE LA INVESTIGACIÓN .....	17

**CAPÍTULO II****MARCO TEÓRICO**

2.1 ANTECEDENTES DE LA INVESTIGACIÓN .....	18
2.2 BASE TEÓRICA.....	20
2.3 MARCO CONCEPTUAL.....	65
2.4 OPERACIONALIZACIÓN DE VARIABLES.....	67

**CAPITULO III****MATERIALES Y MÉTODOS**

3.1 POBLACIÓN Y MUESTRA.....	68
3.2 MÉTODOS.....	71
3.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	72
3.4 MODELO ESPIRAL.....	72
3.5 REQUERIMIENTO MÍNIMO DEL SOFTWARE .....	78

**CAPÍTULO IV****RESULTADOS Y DISCUSIÓN**

4.1 ANÁLISIS Y DISEÑO DEL SISTEMA .....	79
4.2 ESTRUCTURA DE LA BASE DE DATOS .....	88
4.3 IMPLEMENTACIÓN DEL SISTEMA.....	90
<b>CONCLUSIONES .....</b>	<b>103</b>
<b>RECOMENDACIONES .....</b>	<b>104</b>
<b>BIBLIOGRAFÍA .....</b>	<b>105</b>

**INDICE DE DIAGRAMAS**

Diagrama N° 1 de caso de uso asociado al usuario estudiante .....	81
Diagrama N° 2 de caso de uso asociado al usuario estudiante y otro. ....	82
Diagrama N° 3 de caso de uso asociado al usuario bibliotecario.....	83
Diagrama N° 4 de caso de uso asociado al usuario administrador. ....	84
Diagrama N° 5 de contexto del software .....	85
Diagrama 6 de contexto del Software Nivel 1.....	85
Diagrama N° 7 Contexto del Software Nivel 2 “Petición”.....	86
Diagrama N° 8 de contexto del Software Nivel 2 “Registrar” .....	86
Diagrama N° 9 de contexto del Software Nivel 2 “Consultas” .....	87
Diagrama N° 10 de contexto del Software Nivel 2 “Tareas” .....	87
Diagrama N° 11 de contexto del Software Nivel 2“Tareas” .....	88
Diagrama N° 12 Base de datos del Software .....	89

**INDICE DE FIGURAS**

Figura N° 1: Modelo espiral para el proceso del software .....	25
Figura N° 2 Tecnologías agrupadas bajo el concepto de AJAX.....	41
Figura N° 3: Fases para el prototipado de la interfaz .....	44
Figura 4. Diseño de casos de prueba.....	52
Figura N° 5. Prueba de caja negra.....	54
Figura 6. Ejemplo de prueba de caja blanca .....	75
Figura 7 Grafo de flujo de la función Préstamo .....	76
Figura 8. Ventana principal del software .....	90
Figura 9. Ventana para el registro de los usuarios .....	91
Figura 10. Ventana de acceso al software .....	91
Figura 11. Ventana de respuesta de petición de consulta bibliográfica.....	92
Figura 12. Ventana para el llenado de encuestas .....	93
Figura 13. Ventana de opinión y foros.....	93
Figura 14. Ventana para el ingreso de nuevos textos (administrador) .....	94
Figura 15. Ventana para la realización de préstamos (bibliotecario).....	94
Figura 16. Validación W3C del código fuente del software .....	95
Figura 17. Validación W3C del software .....	96

**INDICE DE TABLAS**

Tabla No... 1 Atributos de usabilidad.....	49
Tabla 2 Cálculo de puntos de función para el software.....	57
Tabla 3: preguntas para calcular el valor de complejidad del software. ....	58
Tabla 4. Factores de peso del calificar al software.....	59
Tabla 5 Puntaje que el usuario otorga al software .....	67
Tabla 6. Estudiantes matriculados en la FINESI-UNA-PUNO ciclo 2008-I.....	68
Tabla 7. Número de estudiantes a encuestar por semestre .....	70
Tabla 8. Ejemplo de prueba de caja negra.....	77
Tabla 9. Parámetros de medición en los módulos complejo y mediano .....	97
Tabla 10. Indicadores del Valor de ajuste de complejidad. ....	97
Tabla 11. Factores de medición del peso de complejidad.....	98
Tabla 12. Resultados obtenidos .....	101

## RESUMEN

Las bibliotecas especializadas poseen colecciones ricas sobre un área específica, están diseñadas para responder a necesidades profesionales concretas y ponerlos al alcance de los usuarios, con criterios y métodos de búsqueda eficiente de los contenidos. Actualmente la consulta bibliográfica en la biblioteca especializada de la FINESI-UNA-PUNO es manual y poco eficiente, por lo que obtener información requerida demanda mucho tiempo, además la presencia física del usuario es indispensable. El objetivo de la presente investigación fue implementar un Software de consulta virtual que mejore la atención al usuario en la Biblioteca Especializada de la FINESI-UNA-PUNO. La metodología para la implementación del software consistió en utilizar el modelo espiral, para analizar y diseñar módulos de consulta bibliográfica específicas se efectuó entrevistas y prototipos a los usuarios los que fueron implementados con lenguaje de programación Php, Java Script; y tecnología Ajax. Para medir la aceptación del software fue probado con una muestra de 49 estudiantes, con un 90% de confianza y la prueba estadística Z para proporciones. Las conclusiones fueron: Se logró diseñar el Software de consulta virtual logrando mejorar el servicio de atención al usuario en la Biblioteca especializada de la FINESI-UNA-PUNO donde más del 80 % de usuarios calificó al software como eficiente permitiendo agilizar los préstamos y devoluciones de bibliografía reduciendo el tiempo significativamente. Se implementó módulos de información adicional que permiten facilitar la elección de su consulta con un diseño de interfaz de fácil manipulación.

**Palabras Claves:** Software de consulta, Java Script, Hypertext Pre-processor, Atención al usuario, Tecnologías Ajax

## ABSTRACT

Specialized libraries have collections rich in a specific area, designed to respond to specific professional needs and put them within the reach of users, with criteria and methods of efficient search of the contents. At the moment the bibliographic consultation in the specialized library of the FINESI-A-PUNO is manual and little efficient, reason why obtaining information requires a lot of time, besides the physical presence of the user is indispensable. The objective of the present investigation was to implement a Virtual Consultation Software that improves the attention to the user in the Specialized Library of FINESI-UNA-PUNO. The methodology for the implementation of the software was to use the spiral model, to analyze and design specific bibliographic reference modules, interviews and prototypes were carried out to the users that were implemented with programming language Php, Java Script; And Ajax technology. To measure acceptance of the software was tested with a sample of 49 students, with 90% confidence and the statistical test Z for proportions. The conclusions were: It was possible to design the virtual consultation software to improve the user service in the specialized library of FINESI-UNA-PUNO where more than 80% of users qualify the software as efficient allowing to facilitate the loans and returns of Bibliography reducing time significantly. Additional information modules were implemented to facilitate the choice of your query with an easy-to-use interface design.

**Keywords:** Query software, Java Script, Hypertext Pre-processor, user attention, Ajax Technologies

## INTRODUCCIÓN

El objetivo de la presente investigación fue Implementar un Software de consulta virtual que permita mejorar el servicio de atención al usuario en la biblioteca especializada de la FINESI–UNA-PUNO–2008.

El presente trabajo presenta los siguientes capítulos:

En el Capítulo I se presenta el planteamiento de la investigación, el problema, los objetivos la justificación, los alcances y limitaciones que se presentó. También los objetivos y relevancia de la investigación.

En el Capítulo II se aborda el marco teórico sobre, el Software, la metodología de diseño, además de la definición de término básicos.

En el Capítulo III se presentan los materiales y métodos que fueron usados para la implementación del software. Los procesos que fueron necesarios, tales como el análisis, diseño e implementación de la base de datos, el tratamiento de datos para la validación del software por parte del usuario.

En el capítulo IV se ofrecen los resultados de las pruebas que se efectuaron, la interpretación y la discusión de estos.

Finalmente se presentan las conclusiones, recomendaciones y sugerencias para próximos trabajos.

Para finalizar la tesis se adjunta los anexos.

# CAPÍTULO I

## PLAN DE INVESTIGACIÓN

### 1.1. PLANTEAMIENTO Y DEFINICIÓN DEL PROBLEMA

#### 1.1.1. Planteamiento del problema

En la Universidad Nacional del Altiplano – Puno, se ofertan 40 escuelas profesionales, una de ellas es la Escuela profesional de Ingeniería Estadística e Informática, en donde se automatiza diferentes procesos administrativos.

Actualmente todas las facultades cuentan con una biblioteca especializada, pero pocos tienen un software que permita a los usuarios realizar tareas como consultas, registro, etc. Sin embargo, la facultad de Ingeniería Estadística e Informática (FINESI) que utiliza como herramienta diaria la informática, no cuenta con un software de consultas para la biblioteca especializada que realice, control, consulta y búsqueda de su bibliografía existente, esto conlleva a realizar investigaciones sobre la atención en la biblioteca especializada de la FINESI.

Aplicando las preguntas a los directos encargados de la biblioteca especializada de la FINESI, se logra reunir información tales como: Realizar una serie de gestiones para implementar un software, acumular hardware necesario como computadora, Hub, cables entre otros. Algunas de las bibliotecas especializadas de la UNA – PUNO utilizan el software de la biblioteca central con un filtro de búsqueda de consulta bibliográfica por Facultad y sus características no son de mucha ayuda para los estudiantes de la FINESI, sin dejar de mencionar que la presencia física del usuario es indispensable en el local de la biblioteca especializada.

Actualmente las consultas bibliográficas en la biblioteca especializada de la FINESI es manual, en muchos casos la información requerida demanda mucho tiempo.

Otro escenario común es la poca información del número y clasificación de libros existentes, que obliga al estudiante a consultar todos los libros con el riesgo de no encontrar el tema requerido, tener la duda si aún existe un ejemplar de él o cuantos ejemplares existen en caso de haberse perdido uno de ellos, lo que amerita realizar las consultas en tiempo real.

A los inconvenientes detallados anteriormente, se suman las constantes innovaciones tecnológicas que hacen imprescindible elaborar un software de consultas virtual que se adapten a las necesidades y requerimientos de los usuarios de la Biblioteca Especializada de la FINESI. Además, permita recoger, consultar y tratar la información que mejore la atención de los usuarios.

### **1.1.2. Definición del problema**

Frente a las razones presentadas es necesario plantear alternativas que contribuyan a la solución.

Por lo, que se plantea la siguiente pregunta ¿En qué medida el Software de consulta virtual mejorará el servicio de atención de al usuario en la biblioteca especializada de la FINESI-UNA - PUNO - 2008?

## **1.2. Objetivo general y específicos**

### **1.2.1 Objetivo general**

Implementar un Software de consulta virtual que mejorara la atención al usuario en la biblioteca especializada de la FINESI–UNA-PUNO–2008.

### **1.2.2 Objetivos específicos**

1. Analizar, diseñar e implantar módulos de consultas bibliográficas específicas.
2. Implementar módulos de información adicional que le permita facilitar la elección de su consulta.
3. Implementar módulos que permitan agilizar los préstamos y devoluciones de bibliografía.
4. Diseñar una interfaz que permita manipular con facilidad los módulos del software.

### 1.3. Hipótesis

La implementación del Software de consulta virtual mejorará significativamente el servicio de atención al usuario en la biblioteca especializada de la FINESI-UNA - PUNO – 2008.

### 1.4. Justificación de la investigación

La implementación del software de consulta virtual permitirá mejorar la atención al usuario en la biblioteca especializada de la FINESI–UNA-PUNO–2008. también permitirá recoger, consultar y tratar la información sobre: Control de registro de bibliografía existente, libros prestados, libros devueltos, cantidad de deudores de bibliografía, etc. que ayude al usuario bibliotecario y/o administrador a mejorar el servicio atención en la biblioteca especializada mediante los módulos implantados, en otro contexto el software implementado permita realizar al usuario estudiante, docente y otros consultas bibliográficas virtualmente y que además este software no demande mayores recursos como equipos de cómputo, adaptadores de red, cableado entre otros.

Por lo mencionado, la necesidad de implementar un software orientado a la web es necesario.

Actualmente internet es el medio más utilizado por la población, que se define como un medio de información y de comunicación. Sin duda alguna internet se ha vuelto una herramienta muy útil para todos los sectores que podemos mencionar educación, diseño, industria, etc.,) el cual nos lleva a desarrollar una aplicación para nuestra biblioteca especializada utilizando

tecnología Web, de manera que se pretende lograr su utilización por todos los interesados en nuestra facultad.

La biblioteca especializada (FINESI) en su ambiente contiene textos entre libros de especialidad, folletos, diccionarios, enciclopedias, tesis entre otros, Todos ellos registrados manualmente haciendo tedioso manejar la información requerida, lo que hizo aún más necesario realizar este software, además de ello utilizar la tecnología y beneficios que internet brinda, por lo tanto el “SOFTWARE DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO - 2008” cuenta con módulos implementados que beneficiará principalmente a los usuarios: Bibliotecario, estudiantes, docentes y otros, logrando mejorar así la atención en la biblioteca especializada de la LA FINESI-UNA – PUNO.

### **1.5. Delimitación de la investigación**

El presente trabajo de investigación ha sido desarrollado considerando solo las funciones que se realiza en el local de la biblioteca especializada de la FINESI-UNA- PUNO-2008.

El manejo del software desarrollado considera consultas bibliográficas detalladas con información adicional que ayuda la elección al usuario en su petición (estadística de libros más solicitados, Noticias, encuestas y foros), préstamos, reservas y devolución de bibliografía.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 Antecedentes de la investigación

Existen trabajos relacionados con el uso de Internet

Benito, (1998) En su investigación realizado en la biblioteca central de la Universidad Nacional del Altiplano la inclusión del sistema Micro CDS/ISIS para la organización de la información bibliográfica permitió eliminar los problemas inherentes el manejo manual de esta información, beneficiando al usuario en la rapidez de consulta de bibliografía. Sin embargo, luego de la inclusión del sistema todavía se notaban algunas deficiencias inherentes al aislamiento de la información por parte de este sistema, como la necesidad física de estar presente en el local de la biblioteca, la excesiva carga del sistema en horas punta debido a número de terminales reducido, la no disponibilidad de la información en otros horarios, entre los más principales.

El objetivo de este sistema, era diseñar y construir un sistema de consultas bibliográficas vía Internet.

La metodología asumida para el desarrollo del sistema de consultas bibliográficas vía Internet, fue la de construcción de prototipos, esto porque los requerimientos del usuario final no están completamente definidos, sin embargo, el diseño comprende experiencias de buscadores de información de Internet como Alta vista u otros. Los resultados de este trabajo corroboraron en que Internet, y los servicios que brinda esta, ya van constituyendo el centro de aceptación por parte de la población, además demostrar que problemas de este tipo se pueden resolver con tecnología al alcance.

*Arcaya, (2000)* El propósito de esta tesis fue la tendencia de utilizar Internet con el fin de obtener una mejor relación con el usuario de un sistema informático, trae consigo nuevos retos tanto al desarrollador de aplicaciones como al administrador de la base de datos, pues se tienen que considerar aspectos que van desde la utilización adecuada de una herramienta de desarrollo hasta la configuración de la base de datos. Con el auge que ha tenido el esquema cliente-servidor, han venido también apareciendo una gran cantidad de ambientes que buscan agilizar el proceso de desarrollo de aplicaciones. En Puno nuestra universidad, organizaciones gubernamentales u otras entidades aún no usan soluciones cliente-servidor para resolver sus problemas de carácter informático, y si es que existe es solamente con carácter académico y muy limitado. El objetivo de esta tesis es desarrollar un prototipo de aplicación cliente para consultar actualizaciones de datos académicos en una base de datos remota mediante Internet.

La metodología asumida para el desarrollo del sistema de consultas bibliográficas vía Internet, fue la de construcción de prototipos, esto porque los requerimientos del usuario final no están completamente definidos. Los resultados de este trabajo corroboraron que utilizando la tecnología Internet, se mejoraron las consultas y la actualización de los datos académicos del Instituto Superior Público José Antonio Encinas, se tiene las bases de datos siempre actualizadas para realizar cualquier tipo de consulta y generar los reportes correspondientes. Además, se demostró que las soluciones a través de Internet van causando más aceptación en la solución de los problemas de tipo académico.

## **2.2 Base teórica**

### **2.2.1 Biblioteca<sup>1</sup>**

Una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

### **2.2.2 Sistema de clasificación decimal Dewey<sup>2</sup>**

Como es sabido el sistema de clasificación más usado en España es el de la CDU. La CDU es a la vez un producto del sistema de clasificación creado por Melvil Dewey, un sistema que tiene su origen en 1876 y que acaba de aparecer en su veintigésimoprimera edición.

---

<sup>1</sup> [http://es.wikipedia.org/wiki/Base\\_de\\_datos#Bases\\_de\\_datos\\_est.C3.A1ticas](http://es.wikipedia.org/wiki/Base_de_datos#Bases_de_datos_est.C3.A1ticas)

<sup>2</sup> <http://taranco.eu/cdu/cdu-breve.htm>

La idea nueva de Dewey fue el que el número asignado no indicaba el emplazamiento de los libros en los estantes, sino que respondía a la relación de las materias entre sí.

La otra idea genial de Dewey es hacer que los números asignados a una disciplina sean decimales, por lo que cada nueva cifra que se añade es una subdivisión de la anterior. Ello permite en principio un sinfín de subdivisiones, si es necesario, sin alternar el orden previamente establecido.

### **2.2.3 Software<sup>3</sup>**

Muchas personas asocian el término software con los programas de computadora. Sin embargo, yo prefiero una definición más amplia donde el software no son solo programas, sino todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta. Por lo general, un sistema de software consiste en diversos programas independientes, archivos de configuración que se utilizan para ejecutar estos programas, un sistema de documentación que describe la estructura del sistema, para la documentación para el usuario que explica cómo utilizar el sistema y sitios Web que permitan a los usuarios descargar la información de productos recientes.

### **2.2.4 El proceso de desarrollo de software<sup>4</sup>**

Como el software, al igual que el capital, es el conocimiento incorporado, y puesto que el conocimiento está inicialmente disperso, el desarrollo del software implícito, latente e incompleto en gran medida, es un proceso social

---

<sup>3</sup> Ingeniería del software, séptima edición, Ian Sommerville, capítulo 1, página 5

<sup>4</sup> Pressman Roger S. Ingeniería del Software, España, Concepción Fernández, 2002, Pág. 15

de aprendizaje. El proceso es un diálogo en el que se reúne el conocimiento y se incluye en el software para convertirse en software. El proceso proporciona una interacción entre los usuarios y los diseñadores, entre los usuarios y las herramientas de desarrollo, y entre los diseñadores y las herramientas de desarrollo [tecnología]. Es un proceso interactivo donde la herramienta de desarrollo se usa como medio de comunicación, con cada iteración del diálogo se obtiene mayor conocimiento de las personas involucradas.

### **2.2.5 Fases del proceso de desarrollo de software**

El trabajo que se asocia a la ingeniería del software se puede dividir en tres fases genéricas, con independencia del área de aplicación, tamaño o complejidad del proyecto.

La *fase de definición* se centra sobre el qué. Es decir, durante la definición, el que desarrolla el software intenta identificar qué información ha de ser procesada, qué función y rendimiento se desea, qué comportamiento del sistema, qué interfaces van a ser establecidas, qué restricciones de diseño existen, y qué criterios de validación se necesitan para definir un sistema correcto.

La *fase de desarrollo* se centra en el cómo. Es decir, durante el desarrollo un ingeniero del software intenta definir cómo han de diseñarse las estructuras de datos, cómo ha de implementarse la función dentro de una arquitectura de software, cómo han de implementarse los detalles procedimentales, cómo han de caracterizarse interfaces, cómo ha de traducirse el diseño en un lenguaje de programación (o lenguaje no procedimental) y cómo ha de

realizarse la prueba. Los métodos aplicados durante la fase de desarrollo variarán, aunque las tres tareas específicas técnicas deberían ocurrir siempre: diseño del software, generación de código y prueba del software.

La *fase de mantenimiento* se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. Durante la fase de mantenimiento se encuentran cuatro tipos de cambios:

**Corrección.** Incluso llevando a cabo las mejores actividades de garantía de calidad, es muy probable que el cliente descubra los defectos en el software. El mantenimiento correctivo cambia el software para corregir los defectos.

**Adaptación.** Con el paso del tiempo, es probable que cambie el entorno original (por ejemplo: CPU, el sistema operativo, las reglas de empresa, las características externas de productos) para el que se desarrolló el software. El mantenimiento adaptativo produce modificación en el software para acomodarlo a los cambios de su entorno externo.

**Mejora.** Conforme se utilice el software, el cliente/ usuario puede descubrir funciones adicionales que van a producir beneficios. El mantenimiento perfectivo lleva al software más allá de sus requisitos funcionales originales.

**Prevención.** El software de computadora se deteriora debido al cambio, y por esto el mantenimiento preventivo también llamado reingeniería del software, se debe conducir a permitir que el software sirva para las necesidades de los usuarios finales. En esencia, el mantenimiento preventivo hace cambios en

programas de computadora a fin de que se puedan corregir, adaptar y mejorar más fácilmente.

### **2.2.6 Modelos de procesos de software**

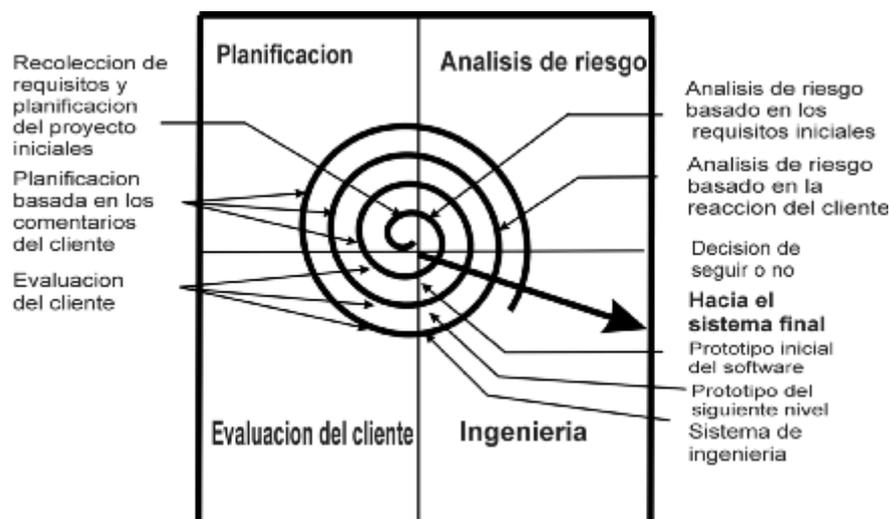
Para resolver los problemas reales de una industria, un ingeniero del software o un equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe al proceso, métodos y capas de herramientas y las fases genéricas. Esta estrategia a menudo se llama modelo de proceso o paradigma de ingeniería del software. Se selecciona un modelo de proceso para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren.

#### **2.2.6.1 El modelo espiral<sup>5</sup>**

El modelo espiral para la ingeniería de software ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico (Análisis, diseño, implementación y validación), como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento: el análisis de riesgo. El modelo representado mediante la espiral de la figura 1, define cuatro actividades principales:

---

<sup>5</sup> Pressman Roger S. Ingeniería del Software, España, Concepción Fernández, 2002, Pág. 24

**Figura N° 1: Modelo espiral para el proceso del software**

(Fuente: <http://es.wikipedia.org>, 2009)

Durante la primera vuelta alrededor de la espiral se definen los objetivos, las alternativas y las restricciones, y se analizan e identifican los riesgos. Si el análisis de riesgo indica que hay una incertidumbre en los requisitos, se puede usar la creación de prototipos en el cuadrante de ingeniería para dar asistencia tanto al encargado de desarrollo como al cliente.

El cliente evalúa el trabajo de ingeniería (cuadrante de evaluación de cliente) y sugiere modificaciones. Sobre la base de los comentarios del cliente se produce la siguiente fase de planificación y de análisis de riesgo. En cada bucle alrededor de la espiral, la culminación del análisis de riesgo resulta en una decisión de "seguir o no seguir".

Con cada iteración alrededor de la espiral (comenzando en el centro y siguiendo hacia el exterior), se construyen sucesivas versiones del software, cada vez más completa y, al final, al propio sistema operacional.

El paradigma del modelo en espiral para la ingeniería de software es actualmente el enfoque más realista para el desarrollo de software y de sistemas a gran escala. Utiliza un enfoque evolutivo para la ingeniería de software, permitiendo al desarrollador y al cliente entender y reaccionar a los riesgos en cada nivel evolutivo. Utiliza la creación de prototipos como un mecanismo de reducción de riesgo, pero, lo que es más importante permite a quien lo desarrolla aplicar el enfoque de creación de prototipos en cualquier etapa de la evolución de prototipos.

### 2.2.7 Metodología Top-down<sup>6</sup>

Top-down es una estrategia de procesamiento de información característica de las ciencias de la información, especialmente en lo relativo al software.

En el modelo Top-down se formula un resumen del sistema, *sin especificar detalles*. Cada parte del sistema se refina diseñando con mayor detalle. Cada parte nueva es entonces redefinida, cada vez con mayor detalle, hasta que la especificación completa es lo suficientemente detallada para validar el modelo. "Top-down" se diseña con frecuencia con la ayuda de "cajas negras" que hacen más fácil cumplir requerimientos, aunque estas cajas negras no expliquen en detalle los componentes individuales.

El enfoque *Top-down enfatiza la planificación y conocimiento completo del sistema*. Se entiende que la codificación no puede comenzar hasta que no se haya alcanzado un nivel de detalle suficiente, al menos en alguna parte del sistema. Esto retrasa las pruebas de las unidades funcionales del sistema hasta que gran parte del diseño se ha completado.

---

<sup>6</sup> <http://es.wikipedia.org/w/index.php?title=Bottom-up&redirect=no>

### 2.2.8 Base de datos<sup>7</sup>

O banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

#### 2.2.8.1 Tipos de bases de datos<sup>8</sup>

##### 1. Según la variabilidad de los datos almacenados

- Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

##### 2. Según el contenido

- Bases de datos bibliográficas

Solo contienen un surrogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica

---

<sup>7</sup> [http://es.wikipedia.org/wiki/Base\\_de\\_datos#Bases\\_de\\_datos\\_est.C3.A1ticas](http://es.wikipedia.org/wiki/Base_de_datos#Bases_de_datos_est.C3.A1ticas)

<sup>8</sup> [http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos)

contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo.

## **2.2.9 Modelo de base de datos**

Un modelo de datos es básicamente una "descripción" de algo conocido como contenedor de datos (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de base de datos; por lo general se refieren a algoritmos, y conceptos matemáticos.

### **2.2.9.1 Base de datos relacional**

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información

puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, Structured Query Language o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

### **2.2.9.2 Componentes de una base de datos<sup>9</sup>**

A continuación, se detallarán las principales características de cada uno de los componentes básicos de una base de datos.

#### **Tablas**

Representan tanto las entidades como las relaciones del modelo E-R (entidad relación) matriz rectangular bidimensional (filas columnas).

Cada elemento o casilla de la matriz es un ítem de dato elemental. Mínimo elemento lógico de acceso y modificación.

Columnas: atributos.

- Nombre único de cada tabla.
- Espacio de valores o dominio.
- Valor nulo: indica valor desconocido o no aplicable.

---

<sup>9</sup> TECSUP, Fundamentos de Base de Datos, Pg. 13

- Captura opcional (puede tomar valores nulos) u obligatoria.
- Su número es a priori fijo.

Filas o tuplas: objetos del mundo real.

- Nunca duplicada.
- Mínima unidad de borrado.
- Registro lógico.
- Su número varía según se realicen operaciones de inserción y borrado en la tabla.
- Pueden estar dispuestas en cualquier orden.

Cardinalidad de una tabla: número de filas.

Nomenclatura: NOMBRE \_TABLA. Nombre\_atributo.

### **Claves**

Necesidad de identificación unívoca y no ambigua.

Clave candidata: agrupación de atributos (quizás uno solo) que identifican sin ambigüedad y de forma unívoca a todas las posibles tuplas de una tabla.

- Mínimo número de atributos: 1, Máximo número de atributos: los que tenga la tabla.
- No debe haber atributos inútiles o superfluos. Si se elimina algún atributo de la clave candidata ya no lo es.

- Siempre hay seguro una clave candidata (la agregación de todos los atributos).

- Puede haber varias claves candidatas.

Clave primaria o principal: se escoge de entre las claves candidatas.

Usualmente se prefiere la de menor tamaño.

Claves alternas o alternativas: el resto de las claves candidatas no escogidas como clave primaria.

Atributo primario: aquel que forma parte de la clave primaria

#### **2.2.10 Normalización de bases de datos<sup>10</sup>**

El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

- \* Evitar la redundancia de los datos.
- \* Evitar problemas de actualización de los datos en las tablas.
- \* Proteger la integridad de los datos.

En el modelo relacional es frecuente llamar tabla a una relación, aunque para que una tabla sea considerada como una relación tiene que cumplir con algunas restricciones:

---

<sup>10</sup> [http://es.wikipedia.org/wiki/Clave\\_for%C3%A1nea#Segunda\\_Forma\\_Normal\\_.282FN.29](http://es.wikipedia.org/wiki/Clave_for%C3%A1nea#Segunda_Forma_Normal_.282FN.29)

- \* Cada columna debe tener su nombre único.
- \* No puede haber dos filas iguales. No se permiten los duplicados.
- \* Todos los datos en una columna deben ser del mismo tipo.

#### **2.2.10.1 Primera forma normal<sup>11</sup>**

La regla de la Primera Forma Normal establece que las columnas repetidas deben eliminarse y colocarse en tablas separadas. Ésta es una regla muy fácil de seguir.

La normalización ayuda a clarificar la base de datos ya organizarla en partes más pequeñas y más fáciles de entender. En lugar de tener que entender una tabla gigantesca y monolítica que tiene muchos diferentes aspectos, usted sólo tiene que entender objetos pequeños y más tangibles, así como las relaciones que guardan con otros objetos también pequeños. No es necesario mencionar que un mejor entendimiento del funcionamiento de su base de datos conducirá a un mejor aprovechamiento de sus activos.

#### **2.2.10.2 Segunda forma normal<sup>12</sup>**

Específicamente: una tabla 1NF está en 2NF si y solo si, dada cualquier clave candidata y cualquier atributo que no sea un constituyente de la clave candidata, el atributo no clave depende de toda la clave candidata en vez de solo una parte de ella.

En términos levemente más formales: una tabla 1NF está en 2NF si y solo si ninguno de sus atributos no-principales son funcionalmente dependientes en

---

<sup>11</sup> [http://www.trucostecnicos.com/trucos/ver.php?id\\_art=278](http://www.trucostecnicos.com/trucos/ver.php?id_art=278)

<sup>12</sup> [http://es.wikipedia.org/wiki/Clave\\_for%C3%83%C2%A1nea#Segunda\\_Forma\\_Normal\\_.282FN.29](http://es.wikipedia.org/wiki/Clave_for%C3%83%C2%A1nea#Segunda_Forma_Normal_.282FN.29)

una parte (subconjunto apropiado) de una clave candidata. (Un atributo no-principal de A es uno que no pertenece a ninguna clave candidata).

Observe que cuando una tabla 1NF no tiene ninguna clave candidata compuesta (claves candidatas consistiendo en más de un atributo), la tabla está automáticamente en 2NF.

### 2.2.10.3 Tercera forma normal<sup>13</sup>

La tercera forma normal (3NF) es una forma normal usada en la normalización de bases de datos. Indica que una tabla está en 3NF si y solo si las dos condiciones siguientes se mantienen:

- ✓ La tabla está en la segunda forma normal (2NF)
- ✓ Ningún atributo no-primario de la tabla es dependiente transitivamente en una clave candidata

### 2.2.11 Internet<sup>14</sup>

Internet es un método de interconexión descentralizada de redes de computadoras implementado en un conjunto de protocolos denominado TCP/IP y garantiza que redes físicas heterogéneas funcionen como una red lógica única, de alcance mundial. Al contrario de lo que se piensa comúnmente, Internet no es sinónimo de World Wide Web (WWW, o "la Web"). Ésta es parte de Internet, siendo uno de los muchos servicios ofertados en la red Internet. La Web es un sistema de información mucho más reciente, El WWW utiliza Internet como medio de transmisión.

<sup>13</sup> [http://es.wikipedia.org/wiki/Clave\\_for%C3%83%C2%A1nea#Tercera\\_Forma\\_Normal\\_.283NF.29](http://es.wikipedia.org/wiki/Clave_for%C3%83%C2%A1nea#Tercera_Forma_Normal_.283NF.29)

<sup>14</sup> <http://es.wikipedia.org/wiki/Internet>

### 2.2.12 Html<sup>15</sup>

El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto)

HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto, algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se debe guardar con la extensión .htm o .html

Estos documentos pueden ser mostrados por los visores o “browsers” de páginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

Sin HTML no existiría el Internet que hoy conocemos, si no se hubiera inventado, seguiríamos con el FTP (Protocolo de Transferencia de Ficheros) mandándonos o bajando archivos que a veces ni sabíamos lo que eran, sin tener la posibilidad de visualizarlo antes. Ahora por medio de una página llena de colores y botones por la cual se puede navegar, se puede tener acceso a bastante información y archivos.

El lenguaje de HTML abrió una puerta al mundo permitiéndoles a las personas expresar sus ideas por medio de páginas y mostrárselas a todas

---

<sup>15</sup> <http://www.monografias.com/trabajos7/html/html.shtml>

las personas de todos los países.

Infinitas son las posibilidades que te brindan las páginas WEB.

Todo el universo de Internet se lo debemos al HTML, ya que todas las páginas con las que se compone la World Wide Web están hechas con el lenguaje de programación HTML.

### **2.2.13 Php**

PHP es un lenguaje de programación usado frecuentemente para la creación de contenido para sitios Web con los cuales se puede programar las páginas html. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones Web muy robustas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de Web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

### **2.2.14 Css**

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) Son un lenguaje formal usado para definir la presentación de un documento

estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.

El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño.

### 2.2.15 ¿Qué es el Consorcio World Wide Web (W3C)?<sup>16</sup>

El Consorcio World Wide Web (W3C) es un consorcio internacional donde las organizaciones miembros, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web.

Diferentes organizaciones, procedentes de diversos puntos del mundo y de campos muy diferentes, forman parte del W3C con intención de participar en un foro neutral para la creación de estándares Web. Los Miembros y un grupo de expertos técnicos, han hecho posible que sea reconocido a nivel internacional por su contribución en el desarrollo de la Web.

Trata de alcanzar su objetivo principalmente a través de la creación de Estándares Web y Pautas. También está involucrado en tareas de educación y difusión, y en el desarrollo de software, sirviendo a su vez como foro abierto de discusión sobre la Web. Para que la Web alcance su máximo potencial, las tecnologías Web más importantes deben ser compatibles entre sí y permitir que cualquier hardware y software, utilizado para acceder a la Web, funcione conjuntamente.

Berners-Lee, que inventó la World Wide Web en 1989, mientras trabajaba en la Organización Europea de Investigación Nuclear (CERN), ha sido el director del W3C desde que se fundó en 1994.

Grupo de Arquitectura Técnica (TAG) del W3C:

John Kemp (Nokia), Larry Masinter (Adobe) y T.V. Raman (Google) Ashok Malhotra (Oracle), Noah Mendelsohn (nombrado, IBM), Jonathan Rees

---

<sup>16</sup> [www.w3c.es](http://www.w3c.es)

(nombrado, Science Commons) y Henry Thompson (Universidad de Edimburgo).

El objetivo de este TAG es llegar a un consenso sobre los principios que rigen la Arquitectura Web e interpretar y aclarar estos principios siempre que sea necesario, así como resolver los problemas relacionados con Arquitectura Web general que se consulten con el TAG, y ayudar a coordinar iniciativas que surjan, dentro o fuera del W3C, para desarrollar la arquitectura a través de varias tecnologías.

### **2.2.16 JavaScript<sup>17</sup>**

Es un lenguaje de programación utilizado para crear programitas encargados de realizar acciones dentro del ámbito de una página Web. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso y tal vez el único con el que cuenta este lenguaje es el propio navegador.

Javascript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante seguido y pensado para hacer las cosas con rapidez a veces con ligereza.

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes. Por un lado, los efectos especiales sobre páginas Web, para

---

<sup>17</sup> Diseño y Programación de páginas Web Editor: Miguel Ángel Pedregosa Pareja.

crear contenidos dinámicos y elementos de la página que tengan movimiento, cambian de color o cualquier otro dinamismo. Por el otro, Javascript nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear paginas interactivas con programas como calculadoras, agendas o tablas de calculo

### 2.2.17 MySql<sup>18</sup>

MySQL es la base de datos open source más popular y, posiblemente, mejor del mundo. Su continuo desarrollo y su creciente popularidad están haciendo de MySQL un competidor cada vez más directo de gigantes en la materia de las bases de datos como Oracle

MySQL es un sistema de administración de bases de datos (Database Management System, DBMS) para bases de datos relacionales. Así, MySQL no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su inter actuación con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

---

<sup>18</sup> <http://www.espestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm> 16-08-2005 Autor: José Manuel Pérez

También es muy destacable, la condición de Open Source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

### **2.2.18 Apache**

El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1[1] y la noción de sitio virtual.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

### **2.2.19 Ajax<sup>19</sup>**

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación Web que estaba apareciendo.

---

<sup>19</sup> <http://www.librosweb.es/ajax/capitulo1.html>

En realidad, el término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se puede traducir como "JavaScript asíncrono + XML".

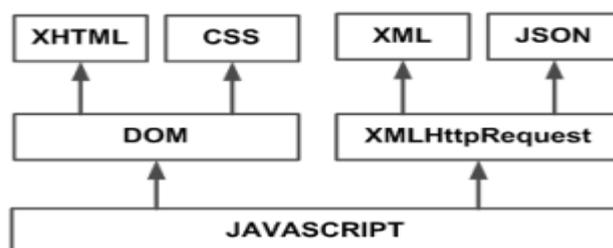
Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- \* XHTML y CSS, para crear una presentación basada en estándares.
- \* DOM, para la interacción y manipulación dinámica de la presentación.
- \* XML, XSLT y JSON, para el intercambio y la manipulación de información.
- \* XMLHttpRequest, para el intercambio asíncrono de información.
- \* JavaScript, para unir todas las demás tecnologías.

Tecnologías agrupadas bajo el concepto de AJAX

**Figura n° 2 Tecnologías agrupadas bajo el concepto de AJAX**



**Fuente:** (Ian Sommerville, 2005)

### 2.2.20 Notepad2

Es editor de texto de código abierto para Microsoft Windows, lanzado bajo la licencia de software GNU GPL. Balmer basó Notepad2 en el principio del bloc de notas de Microsoft: pequeño, rápido y sin "flecós".

Ofrece resaltado de sintaxis para los lenguajes de programación siguientes: ASP, lenguaje ensamblador, C, C++, C#, CGI, CSS, HTML, Java, JavaScript, NSIS, Pascal, Perl, PHP, Python, SQL, VB, VBScript, XHTML y XML. También ofrece resaltado de sintaxis para los formatos de archivos siguientes: BAT, DIFF, INF, INI y REG.

### 2.2.21 MD5<sup>20</sup>

En PHP se utiliza la función MD5 (Message Digest 5), que es una función hash irreversible (de un sólo sentido), es decir, encripta el password tecleado por el usuario y es imposible que partiendo desde la cadena encriptada se vuelva a la contraseña origen. Por esto mismo no hay problema de que alguien pueda acceder al campo encriptado de la base de datos.

Como en la base de datos se guarda la contraseña encriptada, cuando un usuario quiere acceder, habrá que realizar una comparación entre el password que introduce encriptado en MD5, y lo que tenemos en la base de datos, (que es la contraseña encriptada en MD5), si coincide se le permite el acceso, si no, se rechaza.

---

<sup>20</sup> [http://php.ciberaula.com/articulo/encriptacion\\_md5\\_php](http://php.ciberaula.com/articulo/encriptacion_md5_php)

### **2.2.22 El proceso de diseño de la interfaz de usuario<sup>21</sup>**

El diseño de la interfaz de usuario (UI) es un proceso iterativo donde los usuarios interactúan con los diseñadores y prototipos de la interfaz para decidir las características, organización, apariencia y funcionamiento de la interfaz de usuario del sistema. A veces, se construye el prototipo de la interfaz por separado en paralelo con otras actividades de la ingeniería del software. Más comúnmente, en especial cuando se utiliza un desarrollo iterativo, el diseño de la interfaz de usuario se lleva a cabo de forma incremental conforme se desarrolla el software. En ambos casos, sin embargo, antes de que empiece la programación, debe haber desarrollado e, idealmente, probado algunos diseños en papel.

#### **2.2.22.1 Análisis del usuario**

En el proceso de análisis del usuario, se desarrolla una comprensión de las tareas que éste realiza, su entorno de trabajo, los otros sistemas que utiliza, cómo interactúan con el resto de las personas en su trabajo. Etcétera, Para productos con una diversa variedad de usuarios, se debe intentar desarrollar esta comprensión a través de grupos de discusión, pruebas con usuarios potenciales y ejercicios similares.

#### **2.2.22.2 Prototipado del sistema**

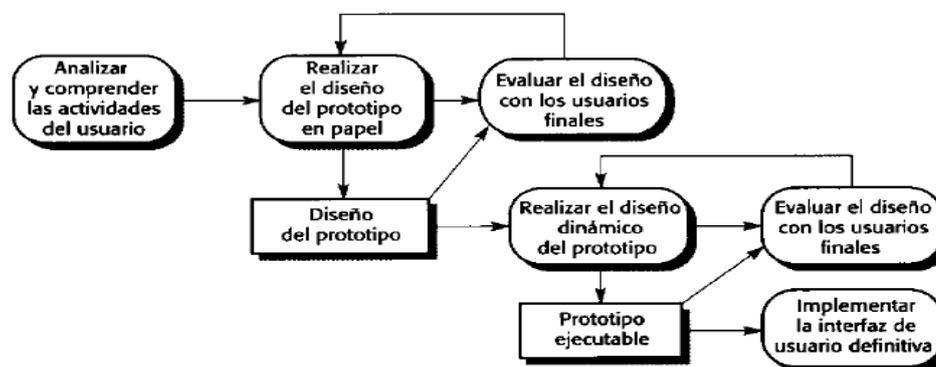
El diseño y desarrollo de la interfaz de usuario es un proceso iterativo, Aunque los usuarios pueden hablar de las facilidades que necesitan de una interfaz, es muy difícil para ellos ser específicos hasta que ven algo tangible.

---

<sup>21</sup> Ingeniería del software, séptima edición, Ian Sommerville, capítulo 16, página 344

Por lo tanto, se deben desarrollar prototipos del sistema y exponerlos a los usuarios, quienes pueden entonces guiar la evolución de la interfaz.

**Figura N° 3: Fases para el prototipado de la interfaz**



Fuente: (Ian Sommerville,2005)

### 2.2.22.3 Evolución de la interfaz

Aunque obviamente se habrá hablado con los usuarios durante el proceso de prototipado, también se debe tener una actividad de evaluación más formalizada donde se recopile información sobre las experiencias reales de los usuarios con la interfaz

... “ Se puede utilizar la construcción de prototipos como parte del proceso de la ingeniería de requerimientos y, en este caso, tiene sentido empezar el proceso de diseño de la UI en esa etapa.” ...

Una actividad crítica del diseño de la UI es el análisis de las actividades del usuario que deben ser soportadas por el sistema informático. Si no se entiende lo que los usuarios quieren hacer con el sistema, no se podrá llevar a cabo un diseño real y eficaz de la interfaz de usuario. Para desarrollar esta

comprensión, puede utilizar técnicas como el análisis de tareas, estudios etnográficos, entrevistas de usuarios y observaciones o, comúnmente, una mezcla de todas ellas.

Un reto para los ingenieros involucrados en el análisis de usuarios es encontrar una forma de describir los análisis de modo que comuniquen la esencia de las tareas a los otros diseñadores y a los usuarios mismos. Notaciones como los diagramas de secuencia de UML pueden describir las interacciones del usuario y son ideales para comunicarse con los ingenieros de software. Sin embargo, otros usuarios pueden pensar que estos diagramas son demasiado técnicos y no intentarán comprenderlos. Debido a que es muy importante implicar a los usuarios en el proceso de diseño, normalmente habrá que desarrollar escenarios en lenguaje natural para describir las actividades del usuario.

#### **2.1.22.4 Técnicas de análisis<sup>22</sup>**

Existen tres técnicas base de análisis del usuario: análisis de tareas, entrevistas y cuestionarios, y etnografía. El análisis de tareas y las entrevistas se centran en el individuo y en su trabajo, mientras que la etnografía adopta una perspectiva más general y considera cómo interactúan las personas, cómo organizan su entorno de trabajo y cómo cooperan para resolver problemas.

Normalmente, se recopila información para el HTA (Análisis de Tareas Jerárquico) a través de la observación y de entrevistas con los usuarios. En este proceso de entrevistas, se puede recopilar parte de esta información

---

<sup>22</sup> Ingeniería del software, séptima edición, Ian Sommerville, capítulo 16, página 346

adicional y registrarla al lado de los análisis de tareas. Cuando se realizan entrevistas para descubrir lo que los usuarios hacen realmente, se deben diseñar las entrevistas de forma que los usuarios puedan proporcionar información que ellos (en vez del entrevistador) piensen que es importante. Esto significa que no hay que ceñirse rígidamente a una lista preparada de preguntas. Más bien, las preguntas deben ser abiertas y animar a los usuarios a que digan por qué hacen las cosas además de lo que realmente hacen.

#### **2.1.22.4.1 Las entrevistas**

Por supuesto, no son sólo una forma de obtener información para el análisis de tareas: son una técnica general para obtener información. Se pueden complementar las entrevistas individuales con entrevistas en grupo o grupos de discusión. La ventaja de utilizar grupos de discusión es que los usuarios se estimulan entre sí para proporcionar información y pueden terminar discutiendo diferentes formas que han desarrollado de utilizar los sistemas.

El análisis de tareas se centra en cómo trabajan los individuos, pero, por supuesto, la mayoría del trabajo es realmente cooperativo. Las personas trabajan juntas para conseguir un objetivo, y los usuarios encuentran difícil discutir cómo se lleva a cabo esta colaboración. Por lo tanto, la observación directa de cómo trabajan y utilizan los sistemas informáticos los usuarios es una importante técnica adicional de análisis del usuario.

#### **2.2.23 Prototipado de la interfaz de usuario**

Debido a la naturaleza dinámica de las interfaces de usuario, las

descripciones textuales y los diagramas no son adecuados para expresar los requerimientos de éstas. El prototipado evolutivo o exploratorio con la implicación de los usuarios finales es la única forma práctica de diseñar y desarrollar interfaces gráficas de usuario para sistemas software. Implicar al usuario en el proceso de diseño y desarrollo es un aspecto fundamental del diseño centrado en el usuario (Normal y Draper, 1986), un criterio de diseño para sistemas interactivos.

El propósito del prototipado es permitir a los usuarios adquirir una experiencia directa con la interfaz. La mayoría de nosotros encuentra difícil pensar de forma abstracta sobre una interfaz de usuario y explicar exactamente qué deseamos. Sin embargo, cuando se nos presentan ejemplos, es fácil identificar las características que nos gustan y las que no.

Idealmente, cuando se está construyendo el prototipo de una interfaz de usuario, se debe adoptar un proceso de prototipado en dos etapas:

1. Al principio del proceso, hay que desarrollar prototipos en papel — maquetas de los diseños de las pantallas— y mostrárselos a los usuarios finales.
2. Entonces, se perfecciona el diseño y se desarrollan prototipos automatizados cada vez más sofisticados, y se ponen a disposición de los usuarios para realizar pruebas y simulación de actividades.

#### **2.2.24 Evaluación de la interfaz de usuario**

La evaluación de la interfaz es el proceso de evaluar la forma en que se utiliza una interfaz y verificar que cumple los requerimientos del usuario. Por

lo tanto, debe ser parte del proceso de verificación y validación de los sistemas software.

De forma ideal, una evaluación se debe llevar a cabo contra una especificación de la usabilidad basada en atributos de la usabilidad, como se muestra en la Tabla 01. Las métricas de estos atributos de usabilidad se pueden idear. Por ejemplo, en una métrica de aprendizaje, se podría enunciar que a un operador familiarizado con las tareas implementadas le debe ser posible utilizar el 80% de la funcionalidad del sistema después de tres horas de formación. Sin embargo, es más común especificar la usabilidad (si es que se especifica del todo) de forma cuantitativa en vez de utilizar métricas. Por lo tanto, el diseñador tiene que utilizar su criterio y experiencia en la evaluación de las interfaces.

La evaluación sistemática del diseño de La interfaz de usuario puede ser un proceso caro que implica a científicos cognoscitivos y diseñadores gráficos. Es posible que se tenga que diseñar y realizar un número estadísticamente importante de experimentos con los usuarios típicos. Se puede necesitar el uso de laboratorios construidos especialmente con equipos de supervisión. Una evaluación de la interfaz de usuario de este tipo es económicamente poco realista para sistemas desarrollados por pequeñas organizaciones con recursos limitados.

Existen varias técnicas menos costosas y sencillas en la evaluación de interfaces que pueden identificar deficiencias específicas en el diseño de interfaces:

1. Cuestionarios que recopilan información de lo que opinan los usuarios de la interfaz.
2. La observación de los usuarios cuando trabajan con el sistema y «piensan en voz alta» de cómo tratan de utilizar el sistema para Llevar a cabo alguna tarea.
3. «instantáneas» de vídeos del uso típico del sistema.
4. La inclusión de código en el software que recopila información de los recursos más utilizados y de los errores más comunes.

Examinar a los usuarios utilizando un cuestionado es una forma relativamente económica de evaluar una interfaz.

#### 2.2.24.1 Atributos de usabilidad

**Tabla No... 1 Atributos de usabilidad**

Atributo	Descripción
Aprendizaje	¿Cuanto tiempo tarda un usuario nuevo en ser productivo con un sistema?
Velocidad de funcionamiento	¿Cómo responde el sistema a las operaciones de trabajo del usuario?
Robustez	¿Qué tolerancia tiene el sistema a los errores del usuario?
Recuperación	¿Cómo se recupera el sistema a los errores del usuario?
Adaptación	¿Está muy bien atado el sistema a un único modelo de trabajo?

**Fuente:** (Ian Sommerville, 2005)

#### Pruebas del software

...”El proceso de pruebas del software tiene dos objetivos distintos:

1. Para demostrar al desarrollador y al cliente que el software satisface sus requerimientos. Para el software a medida, esto significa que debería haber

al menos una prueba para cada requerimiento de los documentos de requerimientos del sistema y del usuario. Para productos de software genéricos, significa que debería haber pruebas para todas las características del sistema que se incorporarán en la entrega del producto. ... Algunos sistemas pueden tener una fase de pruebas de aceptación explícita en la que el cliente comprueba formalmente que el sistema entregado cumple su especificación.

2. Para descubrir defectos en el software en que el comportamiento de éste es incorrecto no deseable o no cumple su especificación. La prueba de defectos está relacionada con la eliminación de todos los tipos de comportamientos del sistema no deseables, tales como caídas del sistema, interacciones no permitidas con otros sistemas, cálculos incorrectos y corrupción de datos.

El primer objetivo conduce a las pruebas de validación, en las que se espera que el sistema funcione correctamente usando un conjunto determinado de casos de prueba que reflejan el uso esperado de aquél. El segundo objetivo conduce a la prueba de defectos, en los que los casos de prueba se diseñan para exponer los defectos. Los casos de prueba pueden ser deliberadamente oscuros y no necesitan reflejar cómo se utiliza normalmente el sistema. Para las pruebas de validación, una prueba con éxito es aquella en la que el sistema funciona correctamente

Las pruebas no pueden demostrar que el software está libre de defectos o que se comportará en todo momento como está especificado. Siempre es posible que una prueba que se haya pasado por alto pueda descubrir

problemas adicionales con el sistema. Como dijo de forma elocuente Edsger Dijkstra una de las primeras figuras líderes en el desarrollo de la ingeniería del software (Dijkstra e al.. 1972), « *las pruebas sólo pueden demostrar la presencia de errores, no su ausencia*».

Por lo tanto, el objetivo de las pruebas del software es convencer a los desarrolladores del sistema y a los clientes de que el software es lo suficientemente bueno para su uso operacional. La prueba es un proceso que intenta proporcionar confianza en el software.

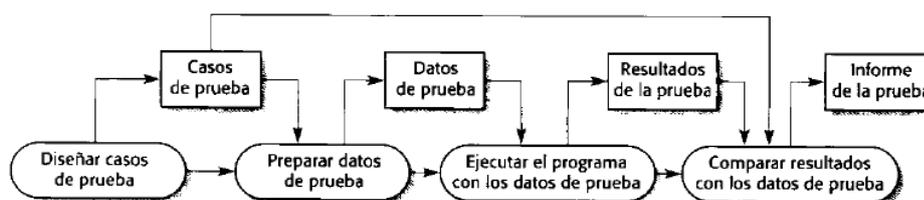
Un modelo general del proceso de pruebas se muestra en la Figura 04. Los casos de prueba son especificaciones de las entradas para la prueba y la salida esperada del sistema más una afirmación de lo que se está probando. Los datos de prueba son las entradas que han sido ideadas para probar el sistema. Los datos de prueba a veces pueden generarse automáticamente. La generación automática de casos de prueba es imposible. Las salidas de las pruebas sólo pueden predecirse por personas que comprenden lo que debería hacer el sistema.

Las pruebas exhaustivas, en las que cada posible secuencia de ejecución del programa es probada, son imposibles. Las pruebas, por lo tanto, tienen que basarse en un subconjunto de posibles casos de prueba. Idealmente, algunas compañías deberían tener políticas para elegir este subconjunto en lugar de dejar esto al equipo de desarrollo, Estas políticas podrían basarse en políticas generales de pruebas, tal como una política en la que todas las sentencias de los programas deberían ejecutarse al menos una vez. De forma alternativa, las políticas de pruebas pueden basarse en la experiencia

de uso del sistema y pueden centrarse en probar las características del sistema operacional. Por ejemplo:

1. Deberían probarse todas las funciones del sistema a las que se accede a través de menús.
2. Deben probarse todas las combinaciones de funciones (por ejemplo. formateado de textos) a las que se accede a través del mismo menú
3. En los puntos del programa en los que el usuario introduce datos, todas las funciones deben probarse con datos correctos e incorrectos.

**Figura 4. Diseño de casos de prueba.**



Fuente: (Roger S. Pressman 1998)

### 2.2.25 Prueba de caja blanca<sup>23</sup>

Es un método de diseño de casos que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca. El ingeniero de software puede obtener casos de prueba que (1) garanticen que se ejecute por lo menos una vez los caminos independientes de cada módulo; (2) ejerciten todas las decisiones lógicas en sus vertiente verdadera y falsa (3) ejecuten todos los bucles en sus límites y

<sup>23</sup> Ingeniería del software, Un enfoque práctico, cuarta edición, Roger S. Pressman, capítulo 16, página 305

con sus límites operacionales, y (4) ejerciten las estructuras internas de los datos para asegurar su validez.

- A menudo creemos que unos caminos lógicos tienen pocas posibilidades de ejecutarse cuando, de hecho, se puede ejecutar de forma normal. El flujo lógico de un programa a veces no es nada intuitivo, lo que significa que nuestras suposiciones intuitivas sobre el flujo de control y los datos nos pueden llevar a tener errores de diseño que solo se descubren cuando comienza la prueba del camino.

- Los errores tipográficos son aleatorios. Cuando se traduce un programa a código fuente en un lenguaje de programación, es muy probable que se den algunos errores de escritura. Muchos serán descubiertos por los mecanismos de comprobación de sintaxis, pero otros permanecerán sin detectar hasta que comience la prueba. Es igual de probable que haya un error tipográfico en un oscuro camino lógico que un camino principal.

#### **2.2.26 Prueba de caja negra<sup>24</sup>**

Se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La prueba de caja negra no es una alternativa a las técnicas de prueba de caja blanca. Más bien se trata de un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca.

---

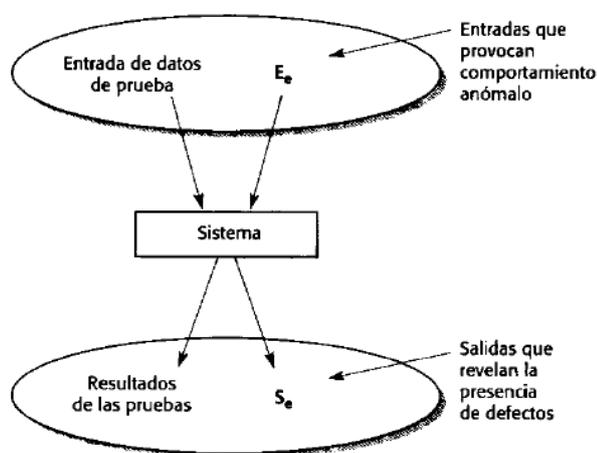
<sup>24</sup> Ingeniería del software, Un enfoque práctico, cuarta edición, Roger S. Pressman, capítulo 16, página 315

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

(1) Funciones incorrectas o ausentes, (2) errores de interfaz, (3) errores en estructuras de datos o en accesos a base de datos externas, (4) errores de rendimiento y (5) errores de de inicialización y de terminación.

A diferencia de la prueba de caja blanca, que se lleva a cabo previamente en el proceso de prueba, la prueba de caja negra tiende a aplicarse durante fases posteriores de la prueba. Ya que la prueba de caja negra ignora intencionalmente la estructura de control, centra su atención en el campo de la información.

**Figura N° 5. Prueba de caja negra.**



**Fuente:** (Roger S. Pressman, 1998)

### 2.2.27 Métrica por puntos de función

Esta métrica se define como una métrica funcional, dado que se enfoca a la funcionalidad que el software proporciona al usuario.

Es una métrica para establecer el tamaño y complejidad de los sistemas informáticos basada en la cantidad de funcionalidad requerida y entregada a los usuarios. Los Puntos de Función miden el tamaño lógico o funcional de los proyectos o aplicaciones de software basado en los requerimientos funcionales del usuario.

...”Partamos de la primera definición para entender las características de la métrica:

- **TAMAÑO.** Es una métrica de tamaño, no de la calidad con la que se hizo ese software, o del valor de ese producto, o del esfuerzo requerido para desarrollarlo, etc.
- **APLICACIONES.** Mide las aplicaciones de software, no considera el hardware que utilizará, ni la administración del proyecto, ni la documentación, etc.
- **FUNCIONALIDAD.** Se refiere a la capacidad del software para que un usuario pueda realizar transacciones (lectura, escritura, etc.) y el guardar datos. Si analizamos a detalle, con estos elementos podemos describir cualquier sistema.
- **USUARIO.** Quien lo va a usar y no quien lo desarrolló o quien lo diseñó.

Así como existe el metro lineal para medir longitudes, Puntos Función es “el metro” para medir tamaño de una aplicación de software. Sus objetivos son:

- Medir lo que el usuario pide y lo que el usuario recibe.

- Medir independientemente de la tecnología utilizada en la implantación del sistema.
- Proporcionar una métrica del tamaño.
- Proporcionar un medio para la estimación del software.
- Proporcionar un factor de normalización para la comparación de distintos softwares.

La técnica de medición del tamaño en punto-función consiste en asignar una cantidad de "puntos" a una aplicación informática según la complejidad de los datos que maneja y de los procesos que realiza sobre ellos, siempre tratando de considerarlo desde el punto de vista del usuario.

Los puntos de función se calculan completando la tabla de la Figura 06. Se determinan cinco características de dominios de información y se proporcionan las cuentas en la posición apropiada de la tabla. Los valores de los dominios de información se definen de la forma siguiente:

**Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

**Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

**Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.

**Número de archivos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).

**Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Una vez que se han recopilado los datos anteriores, a la cuenta se asocia un valor de complejidad. Las organizaciones que utilizan métodos de puntos de función desarrollan criterios para determinar si una entrada en particular es simple, media o compleja. No obstante, la determinación de la complejidad es algo subjetiva.

**Tabla 2 Cálculo de puntos de función para el software.**

Parámetro de Medición	Factor de ponderación					
	Cuenta		Simple	medio	Complejo	
Número de entradas de usuario		x	3	4	5	=
Número de salidas de usuario		x	4	5	7	=
Número de peticiones de usuario		x	3	4	6	=
Número de archivos		x	7	10	15	=
Número de interfaces externas		x	5	7	10	=
Cuenta Total						

**Fuente:** (Roger S. Pressman, 1998)

Para calcular puntos de función, se utiliza la relación siguiente:

$$FP = \text{cuenta-total} \times [0,65 + 0,01 \times \Sigma (Fi) ]$$

Donde cuenta-total es la suma de todas las entradas obtenidas de la figura.

Fi donde i puede ser de uno hasta 14 los valores de ajuste de complejidad

basados en las respuestas a las siguientes preguntas:

**Tabla 3: preguntas para calcular el valor de complejidad del software.**

N	Factor de peso de complejidad	VAC
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?	2
2	¿Se requiere comunicación de datos?	2
3	¿Existen funciones de procesamiento distribuido?	0
4	¿Es crítico el rendimiento?	1
5	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	3
6	¿Requiere el sistema entrada de datos interactiva?	0
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	0
8	¿Se actualizan los archivos maestros de forma interactiva?	4
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?	2
10	¿Es complejo el procesamiento interno?	2
11	¿Se ha diseñado el código para ser reutilizable?	1
12	¿Están incluidas en el diseño la conversión y la instalación?	0
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	4
Total		<b>25</b>

Fuente: (Roger S. Pressman, 1998)

Cada pregunta es respondida usando una escala de rangos que va de 0 a 5

**Tabla 4. Factores de peso del calificar al software**

0	1	2	3	4	5
Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial

**Fuente:** (Roger S. Pressman, 1998)

Los valores constantes de la ecuación (1) y los factores de peso que se aplican a las cuentas de los dominios de información se determinan empíricamente. Una vez que se han calculado los puntos de función, se utilizan de forma análoga a las líneas de código como forma de normalizar las medidas de productividad, calidad y otros atributos del software.

Productividad = PF/persona-mes, Calidad = Errores/ PF, Costo = Dólares / PF

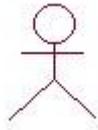
### 2.2.28 Casos de uso

Un diagrama de casos de uso (*Use Case Diagram*) es una representación gráfica de parte o el total de los actores y casos de uso del sistema, incluyendo sus interacciones. Todo sistema tiene como mínimo un diagrama *Main Use Case*, que es una representación gráfica del entorno del sistema (actores) y su funcionalidad principal (casos de uso).

Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno (usuarios u otras aplicaciones).

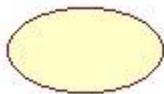
## Elementos

- **Actor:**



Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.

- **Caso de Uso:**



Es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación desde otro caso de uso.

- **Relaciones:**

**Asociación** 

Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

### Dependencia o Instanciación

Es una forma muy particular de relación entre clases, en la cual una clase depende de otra, es decir, se instancia (se crea). Dicha relación se denota con una flecha punteada.

### Generalización

Este tipo de relación es uno de los más utilizados, cumple una doble función dependiendo de su estereotipo, que puede ser de **Uso** (<<uses>>) o de **Herencia** (<<extends>>).

Este tipo de relación está orientado exclusivamente para casos de uso (y no para actores).

**Extends:** Se recomienda utilizar cuando un caso de uso es similar a otro (características).

**Uses:** Se recomienda utilizar cuando se tiene un conjunto de características que son similares en más de un caso de uso y no se desea mantener copiada la descripción de la característica.

### 2.2.29 Indicadores estadísticos<sup>25</sup>

Tamaño de muestra total y su asignación proporcional mediante el MAE para proporciones

---

<sup>25</sup> INEI, Guía práctica construcción e interpretación de indicadores estadísticos, regionales, Lima, Instituto Nacional de Estadística e Informática, 2003. Pág. 15

1) Dado el tamaño de la población  $N$  y los tamaños de los estratos se halla el tamaño de muestra total considerando dos aproximaciones. En la primera aproximación se halló un tamaño de muestra mediante la fórmula:

$$n = \frac{k^2 * p * q * N}{(e^2 * (N-1)) + k^2 * p * q}$$

Donde

**N:** es el tamaño de la población o universo (número total de posibles encuestados).

**k:** es una constante que depende del nivel de confianza que asignemos.

**e:** es el error muestral deseado.

**p:** es la proporción de individuos que poseen en la población la característica de estudio. Este dato es generalmente desconocido y se suele suponer que  $p=q=0.5$  que es la opción más segura.

**q:** es la proporción de individuos que no poseen esa característica, es decir, es  $1-p$ .

**n:** es el tamaño de la muestra (número de encuestas que vamos a hacer).

Si  $f = \frac{n_0}{n} < 0.05$ ,  $n_0$  será nuestro tamaño de muestra definitivo, en caso contrario se reajustará con la fórmula siguiente:

$$n = \frac{n_0}{1 + \frac{(n_0 - 1)}{N}}$$

2) Como la asignación de la muestra debe ser de manera proporcional hay que tomar en cuenta la igualdad de proporcionalidad, la cual especifica que el tamaño de muestra total debe ser proporcional al tamaño de la población y el tamaño de la muestra de cada estrato debe ser proporcional al tamaño de cada estrato, esto es:

$$\frac{n}{N} = \frac{n_h}{N_h}$$

que a su vez da lugar a las siguientes expresiones:

$$\frac{N_h}{N} = \frac{n_h}{n}$$

$$W_h = \frac{n_h}{n}$$

Por tanto, luego de hallar el tamaño de muestra total, se debe asignar los tamaños de muestra a cada estrato mediante la fórmula:

$$n_h = n * \frac{n_h}{N_h} = n * W_h$$

Donde:

$W_h$ : es la ponderación del estrato h.

**Prueba de hipótesis:** Tiene como finalidad corroborar estadísticamente, si la distribución que presentan en base a las observaciones de una muestra están dentro de las especificaciones.

Las pruebas de hipótesis consisten en comprobar, por medio de estadísticos (media muestral, proporción muestra, etc.) obtenidos de una muestra. La

veracidad de una hipótesis llamada hipótesis nula acerca de una distribución o de alguno de sus parámetros. De otra forma se puede decir que la prueba de hipótesis son los métodos estadísticos que llevan a la decisión de aceptar o rechazar una hipótesis. Específicamente una hipótesis nula es una suposición que se hace acerca de un parámetro. El problema consiste en suponer que no existe diferencia entre el parámetro y la información de la muestra a través del estadístico.

Por otro lado, cualquier otra hipótesis diferente que se opone a la hipótesis se llama hipótesis alternativa.

### **Prueba de hipótesis para la proporción poblacional:**

#### **1) Formulación de la hipótesis estadística:**

$$H_0: p \leq P_0$$

$$H_a: p > P_0$$

2) Elección del nivel de significancia. Podemos usar 1%, 5%, 10%

3) Elección de la prueba estadística. Debe seleccionarse aleatoriamente una muestra grande  $n \geq 30$  y entonces la distribución de probabilidad de la proporción muestral es normal.

Para usar la prueba estadística se debe tener en cuenta las siguientes consideraciones

i) Cuando  $P$  es igual o casi igual a 0.5 la distribución de  $p$  es aproximadamente normal tanto para las muestras grandes como muestras pequeñas.

ii) Si  $P$  se aleja de 0.5 la distribución de  $p$  se aproxima a la normal si el tamaño de muestra es grande, de manera tal que se cumpla que  $np$  es igual o mayor que 5

iii) Cuando el tamaño de muestra es pequeño  $n < 30$  o  $p$  está bastante alejado de 0.5 o  $np < 5$ , la distribución binomial o la de Poisson deben ser usadas para encontrar la probabilidad de una prueba de hipótesis.

**Población:** Población: Es el conjunto de sujetos, objetos y procesos que integran o conforman la situación problemática, motivo de la investigación.

**Muestra:** Un subconjunto cualquiera de la población. Para que la muestra nos sirva para extraer conclusiones sobre la población deber ser representativa, **Muestreo:** Procedimiento para la obtención de una muestra

**Afinación Proporcional:** La distribución se hace de acuerdo con el peso (tamaño) de la población en cada estrato.

## 2.3 Marco Conceptual

**2.3.1 Software:** Es la parte lógica conocido como programas de computadoras. Son las instrucciones responsables de que el hardware (la máquina) realice su tarea.

**2.3.2 GenSQL 1.0:** Generadores de código SQL para insertar los datos de los libros digitalizados a la B.D. (creación propia).

**2.3.3 CAT 1.1:** Estructura y organiza un trabajo en HTML que facilita la salida de datos por tema requerido por el usuario (creación propia).

**2.3.4 Biblioteca especializada:** Lugar en donde se encuentra materiales como libros, folletos seleccionados referentes a la escuela profesional.

**2.3.5 Reconocimiento de texto:** Aplicaciones de conversión de imágenes a texto.

**2.3.6 Web:** Cuya base de datos está situada en Internet, datos que pueden ser consultados a través de la red.

**2.3.7 Métrica:** Término que es usado para medir los parámetros referentes al software

**2.3.8 Encuesta:** Las encuestas recogen información de una porción de la población de interés, dependiendo el tamaño de la muestra en el propósito del estudio.

**2.3.9 Consulta virtual:** Consultas realizadas mediante el uso de la Web.

**2.3.10 Módulos del software:** Parte del software que realiza tareas solicitadas por el usuario por ejm. un módulo de consultas bibliográficas.

**2.4 Operacionalización de variables.**

VARIABLES DE ESTUDIO	INDICADOR	INDICE (Categ./Escala)
MEJORA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA-PUNO-2008	Permite realizar consultas bibliográficas específicas (Título, autor, índice área)	No ayuda 0
		Ayuda regular 1
		Si ayuda 2
	Muestra infamación adicional para facilitar la elección de las consultas	No ayuda 0
		Ayuda regular 1
		Si ayuda 2
	Permite agilizar la atención al usuario los módulos de préstamos y devolución de bibliografía	No ayuda 0
		Ayuda regular 1
		Si ayuda 2
	Permita la interfaz manipular con facilidad módulos del software	No ayuda 0
		Ayuda regular 1
		Si ayuda 2

En donde:

**Tabla 5 Puntaje que el usuario otorga al software**

Puntaje que califica el usuario	
0-4	No mejora
5-8	Si mejora

Fuente elaborado por el investigador

## CAPITULO III

### MATERIALES Y MÉTODOS

#### 3.1 Población y muestra

##### 3.1.1 Población:

En el presente trabajo de investigación se consideró a los 344 estudiantes matriculados en FINESI – UNA – PUNO ciclo 2008 – I

**Tabla 6. Estudiantes matriculados en la FINESI-UNA-PUNO ciclo 2008-I**

Escuela profesional	I	II	III	IV	V	VI	VII	VIII	IX	X	TOTAL
Ingeniería estadística e informática	35	37	35	36	37	42	26	28	20	48	344
Total	35	37	35	36	37	42	26	28	20	48	344

**Fuente:** FINESI-UNAP, Nómina de matrícula 2008-I

### 3.1.2 Muestra

#### Muestra y su diseño

La muestra fue conformada por 49 de estudiantes matriculados en el ciclo 2008-I quienes eran usuarios potenciales del servicio de la biblioteca. Dadas las características diferenciales de uso que los estudiantes mostraban hacia el servicio de la biblioteca se consideró determinar el tamaño de muestra simbolizado por  $n$  mediante el método del muestreo aleatorio estratificado donde los estratos fueron los semestres y la población la formaron todos los estudiantes de la FINESI.

Para hallar el tamaño de muestra total se reemplazó en la fórmula con los valores asumidos:

$$n = \frac{k^2 * p * q * N}{(e^2 * (N-1)) + k^2 * p * q}$$

Descripción	Símbolo	valores
Tamaño de muestra sin ajustar	$n$	57
Nivel de confianza	$k$	1.65
Proporción de usuarios que considera el software mejora el servicio en la biblioteca especializada	$p$	0.5
Proporción de usuarios que considera el software no mejora el servicio en la biblioteca especializada	$q$	0.5
Población	$N$	344
error muestral deseado	$e$	0.1

Reemplazando valores se obtuvo los siguientes resultados.

$$n = \frac{1.65^2 * 0.5 * 0.5 * 344}{(0.1^2 * (344 - 1)) + 1.65^2 * 0.5 * 0.5} = 57$$

Como  $n / N$  es 0.1655 es  $>0.05$  se reajusto el tamaño de muestra como sigue:

$$n = \frac{n_0}{1 + \frac{n_0 - 1}{N}}$$

$$n_0 = 57$$

$$n = \frac{57}{1 + (1 + \frac{57}{344})} = 48.86$$

Los cálculos realizados se dan en la siguiente tabla:

**Tabla 7. Número de estudiantes a encuestar por semestre**

SEMESTRES	Nh (Número de alumnos matriculados por semestre)	N (Población)	n <sub>h</sub> Tamaño de muestra ajustado	nh (tamaño de muestra por semestre) n
Semestre I	35	344	49	5
Semestre II	37	344	49	5
Semestre III	35	344	49	5
Semestre IV	36	344	49	5
Semestre V	37	344	49	5
Semestre VI	42	344	49	6
Semestre VII	26	344	49	4
Semestre VIII	28	344	49	4
Semestre IX	20	344	49	3
Semestre X	48	344	49	7
<b>Total</b>	344			49

Fuente: Elaborado por el investigador

n<sub>h</sub>: tiene la siguiente afijación proporcional:  $n_h = \frac{N_h}{N} (n)$

Determinar el número de estudiantes a encuestar por semestre.

(Ejm. Para el primer semestre será  $n_h = 35 \cdot 49 / 344 = 4.98 = 5$ )

## 3.2 Métodos

### 3.2.1 Método de recolección de datos

Se empleó el método de la encuesta y el instrumento fue el cuestionario elaborado para los estudiantes de la FINESI UNA PUNO y personal que labora en la biblioteca especializada de la siguiente manera.

Para el diseño del software se obtuvo los datos mediante entrevistas directas con el personal que labora en la biblioteca especializada y estudiantes presentes en la biblioteca especializada.

#### 3.2.1.1 Entrevistas

Las entrevistas por supuesto, no son solo una forma de obtener información, son una técnica para obtener información. Se complementaron las entrevistas individuales con entrevistas en grupo de discusión. La ventaja de utilizar grupos de discusión es que los usuarios se estimulan entre sí para proporcionar información y pueden terminar discutiendo logrando descubrir así lo que los usuarios hacen realmente de manera que pueden proveer información que ellos piensan que es importante.

#### 3.2.1.2 Encuesta

Este método de recopilación de datos para fines de prueba del software, se realizaron una serie de preguntas mediante un cuestionario realizado para los usuarios.

### **3.3 Metodología de desarrollo de software**

En el presente trabajo de investigación se utilizó la metodología TOP-DOWN es decir se vio en un principio en forma general el diseño del software, diseñándose en un principio diagrama de contexto desde un punto de vista general a lo específico, la presencia de prototipos y entrevistas fueron primordiales para el diseño de la interfaz y recolección de requerimientos.

#### **3.3.1 Diseño de interfaz de software**

El prototipado de interfaz de software se utilizó para mejorar el análisis y diseño se consultó con usuarios (estudiantes de la FINESI), sobre que rostro (Interfaz) debería de tener el software de biblioteca. Aunque los usuarios pueden hablar de las facilidades que necesitan de una interfaz, es muy difícil para ellos ser específicos hasta que ven algo tangible. Por lo tanto, se desarrolló prototipos del sistema para exponerlos a los usuarios, quienes guiaron la evolución de la interfaz para finalmente plasmarlo en un diagrama de contexto con todos los requerimientos definidos por él.

### **3.4 Modelo Espiral**

#### **3.4.1 Análisis y Diseño**

Una vez creado el diagrama de contexto del software el siguiente paso es elegir una estrategia de desarrollo que acompañe al proceso de aplicar ciertas técnicas y principios con el propósito de definir un dispositivo y métodos. Ajustándose mejor según la naturaleza del proyecto, la aplicación,

los métodos, las herramientas a utilizarse, los controles y entregas que se requieren este es el modelo espiral.

### **3.4.2 Implementación**

Para la implementación del software se utilizó el lenguaje de programación PHP Script Language Version 5.2.3, Javascript y la base de datos en MySQL 5.0

### **3.4.3 Evaluación y Pruebas del Software**

#### **3.4.3.1 Pruebas del software**

Una fase de pruebas de aceptación en la que el usuario (estudiante) comprueba formalmente que el sistema entregado cumple su especificación.

#### **3.4.3.2 Prueba de caja blanca**

Para fines de ejemplificar las pruebas de caja blanca se tomó como muestra el módulo préstamos de libros y se tomó una de las funciones llamada préstamo.

Descripción del algoritmo de la figura 07

- Se obtiene un enlace hacia la base de datos
- Obtenemos datos del libro a prestar
- Verificamos si el libro existe
- Extraemos los datos del usuario
- Revisamos los libros que el usuario se presto

- Verificamos si ya se prestó un libro similar al que está pidiendo
- Verificamos si realizo más de 2 libros
- Enviamos un mensaje
- Si se prestó menos de 2
- Obtenemos Cuantos libros de ese código ya fueron prestados
- Obtenemos cuantos fueron reservados
- Verificamos si stock menos estas dos consultas es mayor a cero
- Agregamos un nuevo préstamo a la tabla
- Y quitamos la reserva en caso de que lo haya reservado con anterioridad
- Verificamos si este libro pertenece a los más solicitados
- Si no pertenece agregamos un nuevo registro a la Tabla Más Solicitados
- Si ya pertenece incrementamos su puntuación
- Si no hay stock suficiente se muestra un mensaje
- Si el usuario ya se prestó el mismo libro enviamos un mensaje
- Si el código de libro no existe se envía un mensaje
- Se cierra la conexión a la base de datos

Figura 6. Ejemplo de prueba de caja blanca

```

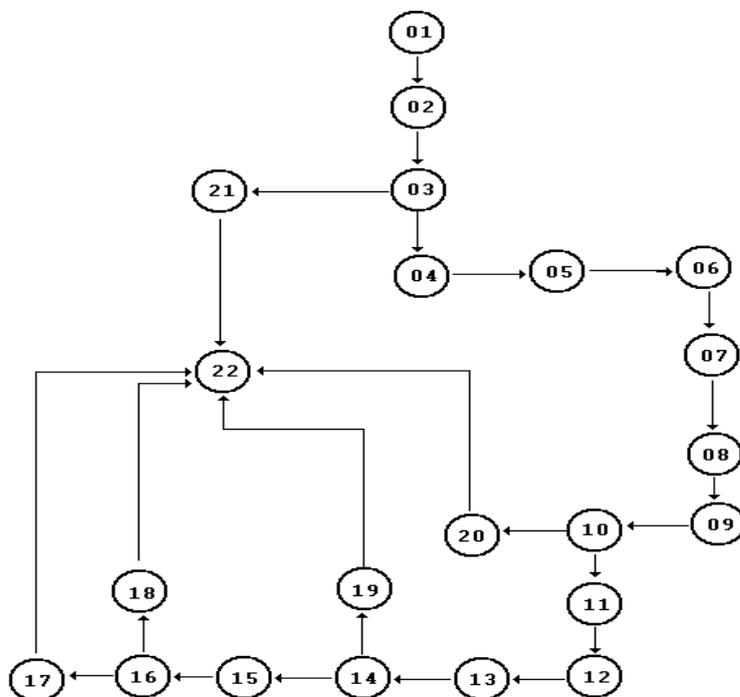
function prestamo($CodUsuario,$CodLibro)
01 {
    $enlace=conectarafinesi();
02 $DatosLibro=rev_cod_libro($enlace,$CodLibro);
03 if($DatosLibro=="")
    {
04     $DatosUsuario=prestamos_usuario($enlace,$CodUsuario);
        $n=0;
05     while($consulta=mysql_fetch_array($DatosUsuario))
        {
06         $n++;
07         if($consulta['codlibro']==$CodLibro)
08             $s=1;
09     }
10     if($s!=1)
11     {
        if($n>=2)
12     {
            echo utf8_encode("Máximo de libros excedido.");
        }
        else
13     {
            $n=num_prestamos($enlace,$CodLibro);
            $m=num_reservados($enlace,$CodLibro);
            $a=num_reservados_u($enlace,$CodLibro,$CodUsuario);
14             if(($DatosLibro['stock']-$n-$m+$a)>0)
            {
                $fecha=sacarfecha();
                $Codprestamo=cod_registro($enlace,"prestamos","codprestamo");
                $Consulta="insert into prestamos values('$CodUsuarioodprestamo','$CodUsuario','$CodLibro','$fecha','N')";
                $Consulta=mysql_query($CodUsuarioonsulta,$enlace);
15                 $Consulta="update reservados set valido='N' where codusuario='$CodUsuario' and codlibro='$CodLibro' and valido='S'";
                $Consulta=mysql_query($CodUsuarioonsulta,$enlace);
                $Consulta="select *from masconsultados where codlibro='$CodLibro'";
                $Consulta=mysql_query($CodUsuarioonsulta,$enlace);
                $DatosLibro=mysql_fetch_array($consulta);
16                 if($DatosLibro=="")
                {
17                     $CodUsuarioonsulta="insert into masconsultados values('$CodLibro','1')";
                    $CodUsuarioonsulta=mysql_query($CodUsuarioonsulta,$enlace);
                }
                else
                {
                    $nlibros=$vari['cantidad'];
                    $nlibros++;
18                     $Consulta="update masconsultados set cantidad='$nlibros' where codlibro='$CodLibro'";
                    $Coonsulta=mysql_query($CodUsuarioonsulta,$enlace);}
                    echo utf8_encode("Prestamo realizado con éxito");
                }
            }
            else
19             echo utf8_encode("no hay libros disponibles.");
        }
    }
    else
20     echo utf8_encode("El libro ya lo a prestado al mismo estudiante.");
    }
    else
21     echo utf8_encode("El código del libro no existe.");
22     mysql_close($enlace);
}

```

Fuente: Elaboración propia

**Figura 7 Grafo de flujo de la función Préstamo**

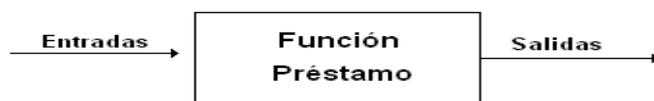
Representación del algoritmo de la figura 06 utilizando grafos.



**Fuente:** Elaboración propia

### 3.4.3.3 Prueba de caja negra

Para esta prueba del software se tomó el módulo préstamo de libros del usuario bibliotecario (debemos recordar que la prueba de caja negra se basa en la funcionalidad del software es decir si el resultado que me proporcionará será el que se espera que haga).



**Tabla 8. Ejemplo de prueba de caja negra**

Caso	Código de libro	Identificador de usuario (ID)	Resultado
Caso de préstamo de un libro	600.EDP.T01	8	Préstamo realizado con éxito
Caso en que se solicite el mismo libro	600.EDP.T01	8	El libro ya se ha prestado al mismo estudiante
Caso en que ya cumplió el máximo de solicitud de préstamo	600.EDP.L01	8	Máximo de libros excedido
Caso en que ya no se dispone de otro ejemplar del libro	620.PLN.B04	10	No hay libros disponibles.
Caso en que el código del libro se haya ingresado erróneamente	-890.PLN.B04	10	El código del libro no existe.

**Fuente:** Elaboración propia

#### 3.4.4 Hardware

- Cámara digital 8.2 MP. (Para el siguiente trabajo se extrajo información de índices de libros de la biblioteca especializada de la FINESI).
- Equipo computacional (Pentium III, 128MB RAM, 1GB Procesador)
- Dispositivo de almacenamiento externo (USB)

#### 3.4.5 Software

- PHP Lenguaje Versión 5.2.3
- Base de datos MySql 5.0
- Apache Web Server 2.2
- Javascript
- Hojas de estilo CSS

- Editor de texto NotPad2

### **3.4.6 Aplicaciones de apoyo**

- Reconocimiento de texto
- GenSQL 1.0: Detallado en el marco conceptual.
- CAT 1.1: Detallado en el marco conceptual.

### **3.5 Requerimiento mínimo del software**

- Equipo computacional (Pentium I ó superior, 128MB RAM, 1GB Procesador)
- Mouse y teclado estándar
- Sistema operativo Melenium o superior
- Tarjeta de Red.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1 Análisis y diseño del sistema

La implementación del SOFTWARE DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO - 2008, tiene por objetivo implementar de consulta virtual que permita mejorar el servicio de atención al usuario.

En seguida se detalla el conjunto de requisitos encontrados, los cuales deben de ser resueltos por el Software de consulta virtual para la atención al usuario en la Biblioteca Especializada.

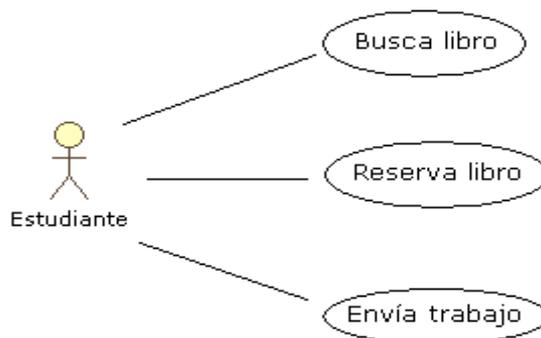
1. Se desea desarrollar la capacidad de consultar y tratar información existente de la biblioteca especializada de la FINESI donde se encuentre a través de la Web.
2. Cuando el usuario acceda al software debe visualizar un entorno que le permita familiarizarse con él.

3. Debe ofrecer al usuario seleccionar el área o áreas que requiera de manera fácil y rápida (Consulta virtual, reservas, devoluciones, consultas bibliográficas entre otros)
4. Debe ofrecer al usuario visualizar información amplia para facilitarle la elección de la tarea que desea hacer como pedir información del texto a solicitar, reservar ó devolver.
5. Debe ofrecer al usuario acceder a otros atributos adicionales como ver encuestas, enviar opiniones, enviar mensajes, y reservar vía Internet.
6. Debe ofrecer al usuario tener control sobre préstamos y devoluciones de bibliografías realizadas diariamente (encargado de la biblioteca especializada)
7. Debe ofrecer al usuario, agregar, modificar los textos existentes en la biblioteca especializada de la FINESI – UNA – PUNO
8. Debe ofrecer al usuario la publicación de encuestas, noticias y los trabajos realizados por los estudiantes (Administrador de biblioteca especializada)
9. Debe ofrecer al usuario la decisión de restringir o no a los usuarios que actúan maliciosamente (Administrado o bibliotecario)
10. Debe ofrecer al usuario reportes de información cuando lo requiera

## Casos de uso asociados con el usuario

En los siguientes diagramas muestra los casos de uso asociados a los usuarios: Estudiante, bibliotecario y administrador del software.

### Diagrama N° 1 de caso de uso asociado al usuario estudiante



**Fuente:** Elaboración propia

**Busca libro:** Aquí el usuario en su modalidad de estudiante o invitado estudiante realiza la búsqueda del libro que desea mediante las opciones por: Título, autor, tema (ver el índice del libro), área (Estadística, Informática, matemáticas, otros y tesis), código, también búsquedas avanzadas como:

Mostrar todos los libros detallados

Buscar solo los libros más solicitados

Buscar solo libros que estén digitalizados  y

Libros más solicitados.

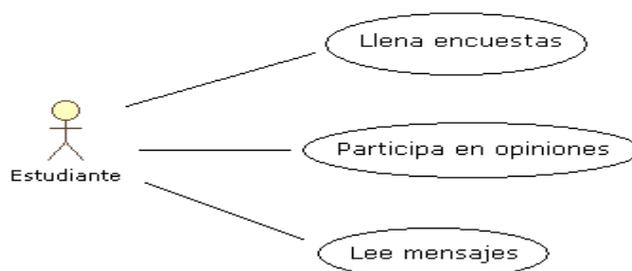
**Reserva libro:** Aquí el usuario en su modalidad de estudiante puede realizar la reserva vía Internet, la reserva realizada está sujeta a un límite de tiempo dado, de no concretar la reserva con el retiro del libro del ambiente de la biblioteca especializada de la FINESI UNA, el software automáticamente

acumulara una falta que pueden ser sancionados (serán determinadas por la administración y/o bibliotecario), de reincidir el usuario con estas faltas, le restringirá el uso del sistema.

Los usuarios que están registrados como invitados solo podrán realizar consultas de textos y están restringidos de usar los demás módulos.

**Enviar Trabajo:** Aquí el usuario estudiante puede enviar su trabajo, siendo enviado en primer lugar al administrador, quien será el encargado de verificar el contenido del libro para luego ser publicado y visto por todos los que accedan al software.

#### Diagrama N° 2 de caso de uso asociado al usuario estudiante y otro.



**Fuente:** Elaboración propia

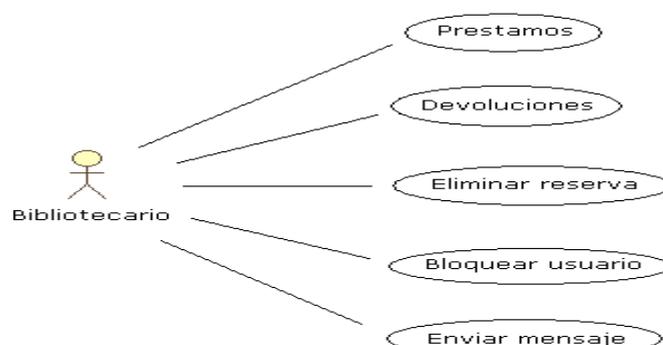
**Llenar encuestas:** Aquí el usuario estudiante puede participar en el llenado de las encuestas que serán publicados por el administrador, cada encuesta será llenado una sola vez por usuario sin opción a corregirlos una vez enviado. Las encuestas son anónimas.

**Participa de opiniones:** Aquí el usuario estudiante participa en opiniones de temas que son creados en el link opiniones. Por ejemplo, alguien podría

solicitar ayuda sobre algún problema que tuvo con un tema específico.

(Foros creados solo con estudiante registrados de modalidad estudiante)

### Diagrama N° 3 de caso de uso asociado al usuario bibliotecario



**Fuente:** Elaboración propia

**Préstamos:** Aquí el usuario bibliotecario puede efectuar los préstamos, primero digita el código del estudiante, y luego el código del libro a prestar.

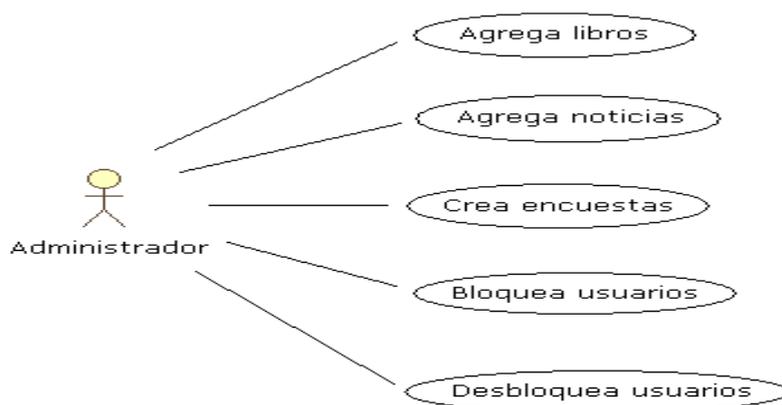
**Devoluciones:** Aquí el usuario bibliotecario al recibir el libro devuelto, primero digita el código del estudiante, y luego el código del libro a devolver registrándolo como libro devuelto.

**Eliminar reserva.** Aquí el usuario bibliotecario también puede eliminar la reserva si lo cree por conveniente solo digitara el código del estudiante y para eliminar ingresa a la opción ver reservados.

**Bloquear usuario:** Aquí el usuario bibliotecario puede bloquear al usuario que haya cometido faltas por ejemplo a los estudiantes que hayan tenido en su poder a textos por más días de los permitidos; en el módulo Adicionar Observación, Digita el código del usuario y detalla el motivo del bloqueo.

**Enviar mensaje:** Aquí el usuario bibliotecario puede enviar mensajes que crea por conveniente a los usuarios puede ser a un usuario que tiene el libro por más días de los permitidos.

**Diagrama N° 4 de caso de uso asociado al usuario administrador.**



Fuente: Elaboración propia

**Agregar libro:** Aquí el usuario Administrador realiza las actualizaciones de los nuevos libros que lleguen a la biblioteca especializada de la FINESI - UNA - PUNO.

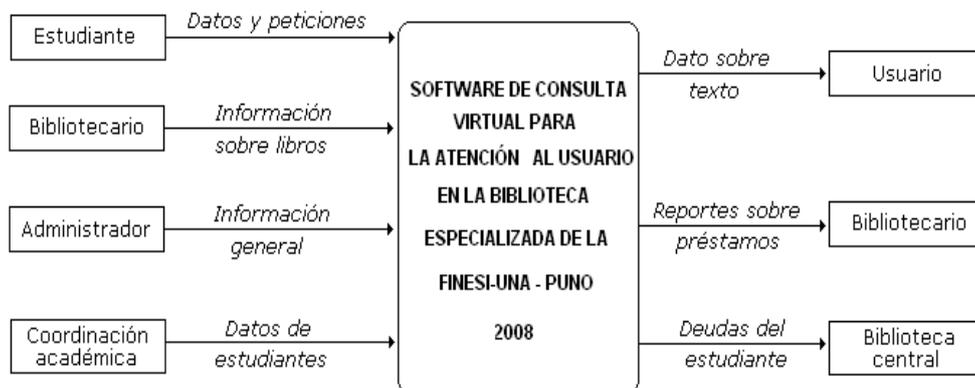
**Agregar noticias:** Aquí el usuario Administrador agregara las noticias que crea por conveniente.

**Bloquea usuario:** Aquí el usuario Administrador puede bloquear al usuario que haya cometido faltas.

**Desbloquea usuario:** Aquí el usuario Administrador puede desbloquear al usuario que le haya solicitado.

**Diagrama N° 5 de contexto del software**

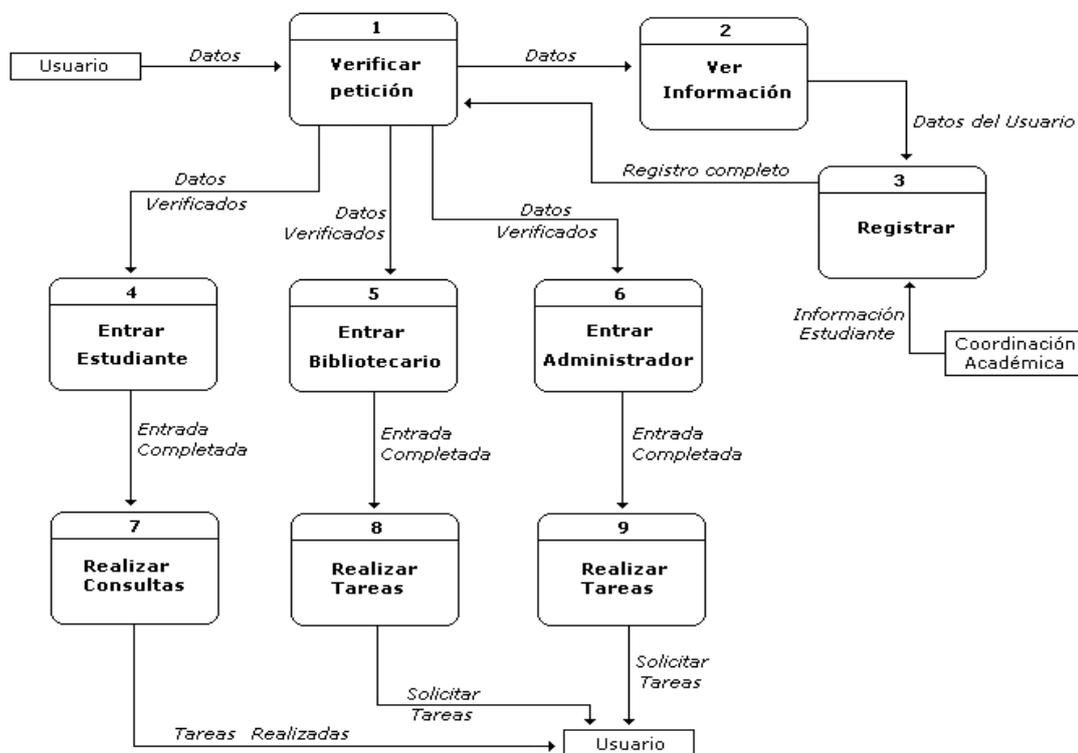
**DIAGRAMA DE CONTEXTO DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO - 2008. NIVEL 0**



Fuente: Elaboración propia

**Diagrama 6 de contexto del Software Nivel 1**

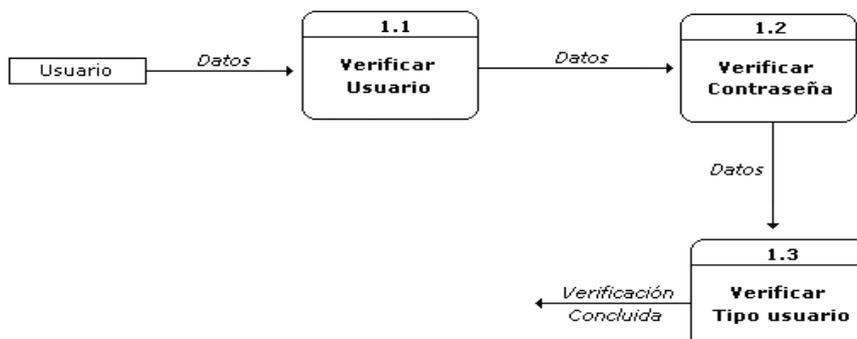
**Diagram general Nivel 1**



Fuente: Elaboración propia

**Diagrama N° 7 Contexto del Software Nivel 2 “Petición”**

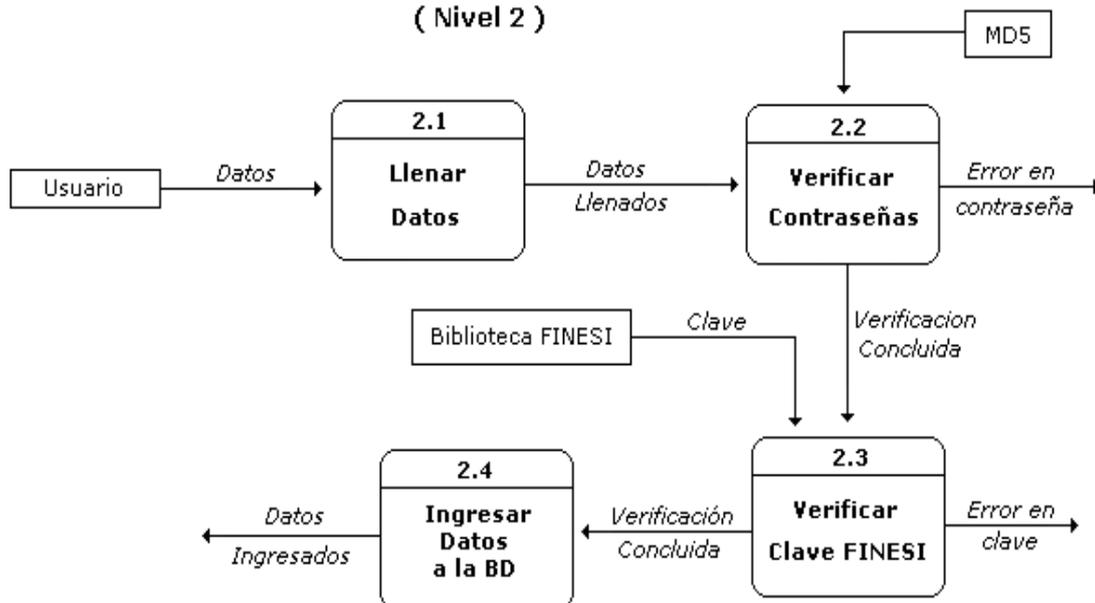
**Diagrama de Sub-Nivel para el proceso "Verificar Petición"**  
( Nivel 2 )



Fuente: Elaboración propia

**Diagrama N° 8 de contexto del Software Nivel 2 “Registrar”**

**Diagrama de Sub-Nivel para el proceso "Registrar"**  
( Nivel 2 )

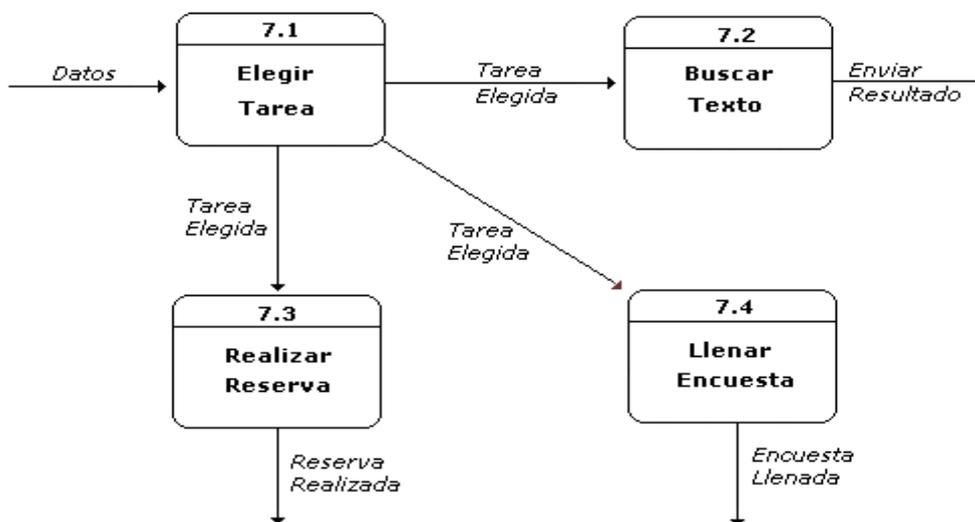


Fuente: Elaboración propia

**Diagrama N° 9 de contexto del Software Nivel 2 “Consultas”**

**Diagrama de Sub-Nivel para el proceso "Realizar Consultas"**

( Nivel 2 )

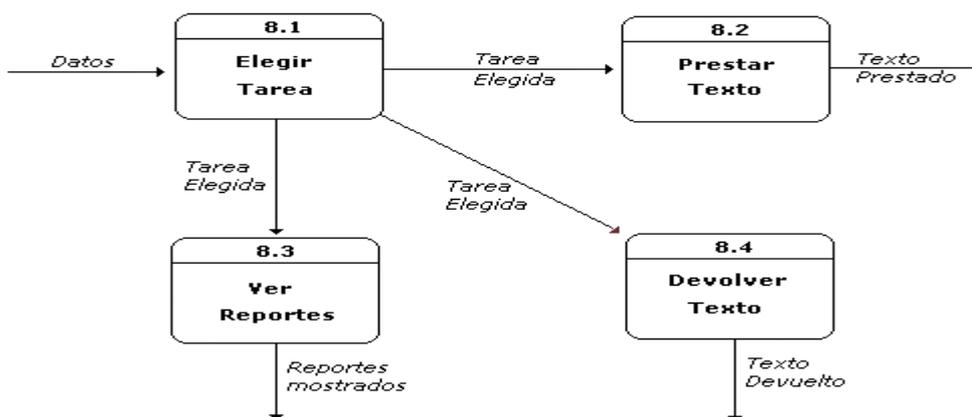


Fuente: Elaboración propia

**Diagrama N° 10 de contexto del Software Nivel 2 “Tareas”**

**Diagrama de Sub-Nivel para el proceso "Realizar Tareas "**

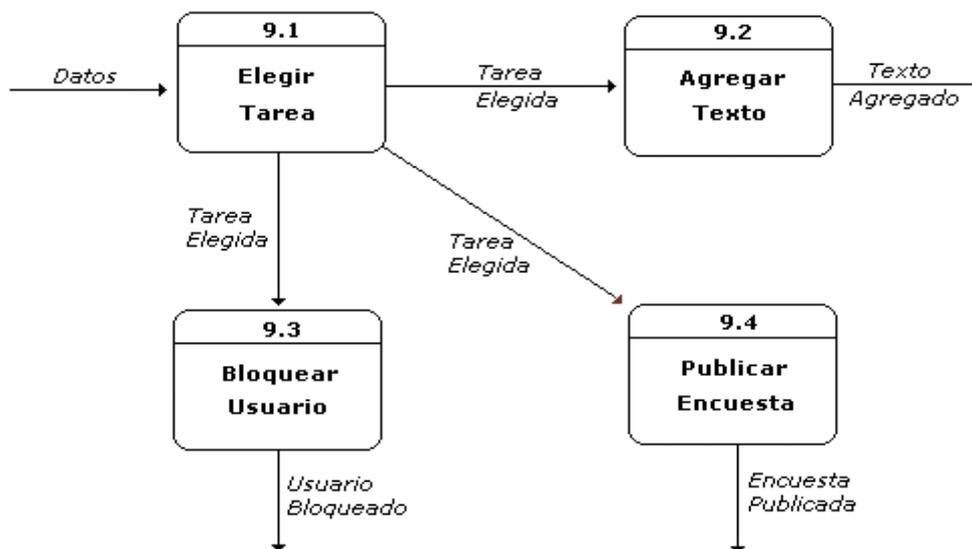
( Nivel 2 )



Fuente: Elaboración propia

**Diagrama N° 11 de contexto del Software Nivel 2“Tareas”**

**Diagrama de Sub-Nivel para el proceso "Realizar Tareas "**  
( Nivel 2 )

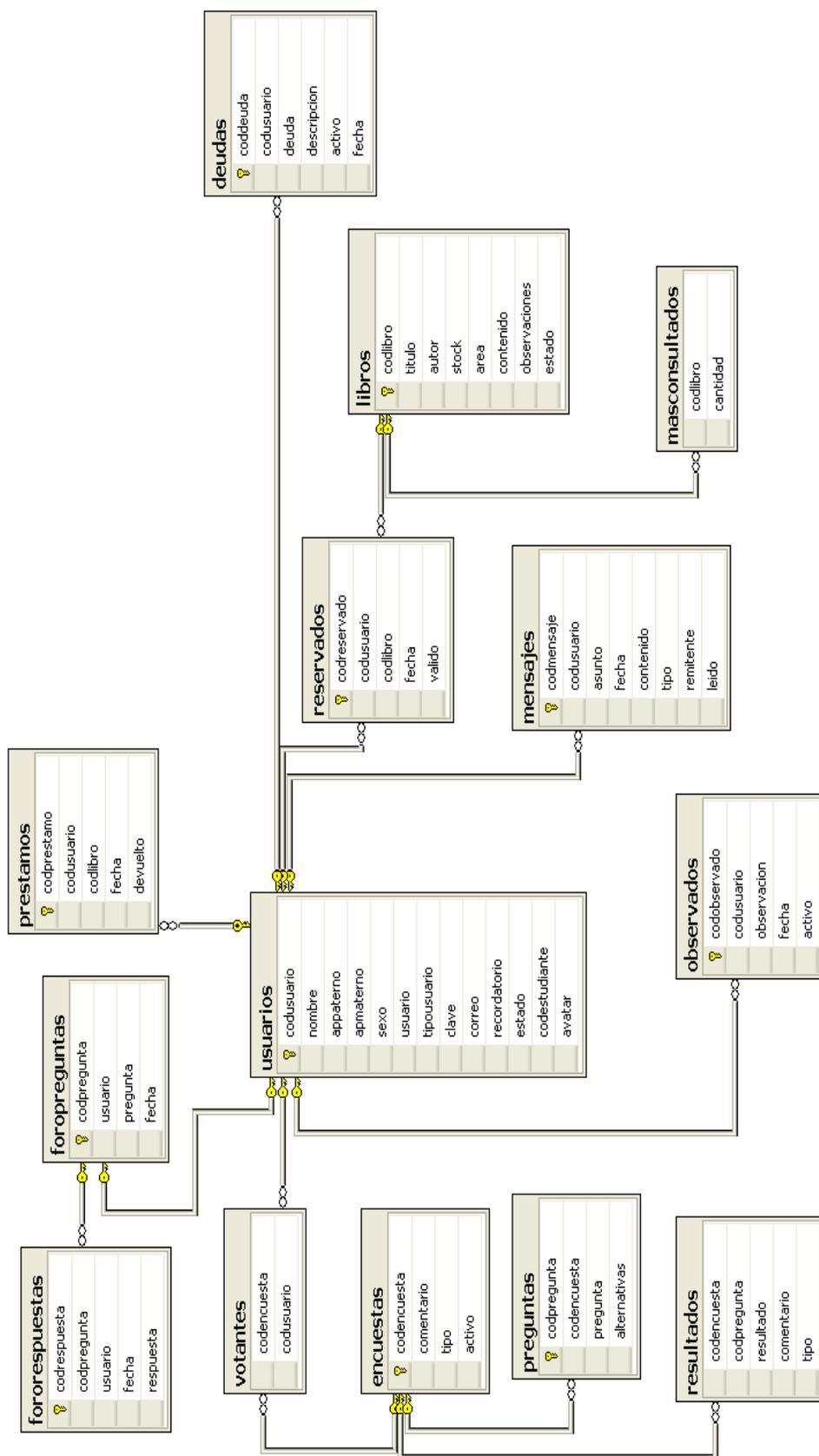


Fuente: Elaboración propia

**4.2 Estructura de la Base de datos**

Luego de realizar de normalizaciones de los datos que se manejaran se llegó a diseñar la base de datos final

Diagrama Nº 12 Base de datos del Software



Fuente: Elaboración propia

### 4.3 Implementación del sistema

Las funciones y procesos fueron implementados con Lenguaje de programación gratuitos Php, Java Script; CSS (hojas de estilo para facilitar el formato de las páginas) y tecnología Ajax para mejorar el rendimiento y disminuir el tiempo de conexión entre el usuario y el servidor sin necesidad de adquirir licencias.

#### Ventana principal

Figura 8. Ventana principal del software



Fuente: Elaboración propia

Ventana para registrar usuarios

Figura 9. Ventana para el registro de los usuarios

Fuente: Elaboración propia

Ventana de acceso a software

Figura 10. Ventana de acceso al software

Fuente: Elaboración propia

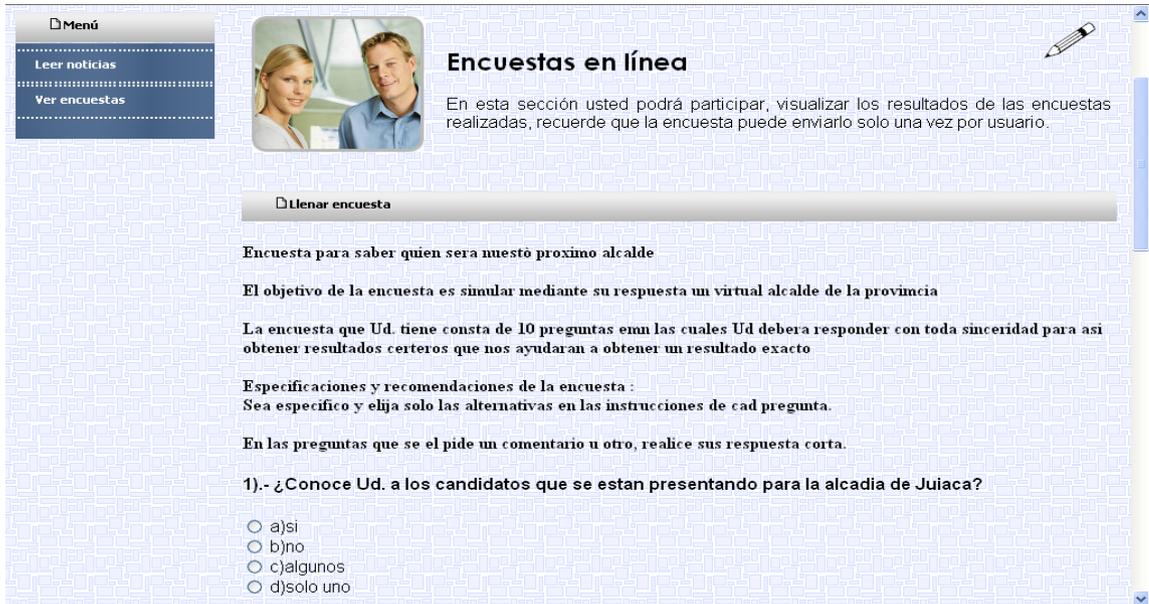
Ventana de consultas bibliográficas

Figura 11. Ventana de respuesta de petición de consulta bibliográfica



Fuente: Elaboración propia

Figura 12. Ventana para el llenado de encuestas



Fuente: Elaboración propia

### Ventana de opiniones – foros

Figura 13. Ventana de opinión y foros



Fuente: Elaboración propia

Ventana de actualización de textos destinado para uso del administrador

Figura 14. Ventana para el ingreso de nuevos textos (administrador)



Fuente: Elaboración propia

Ventana de préstamos destinado para uso del bibliotecario

Figura 15. Ventana para la realización de préstamos (bibliotecario)



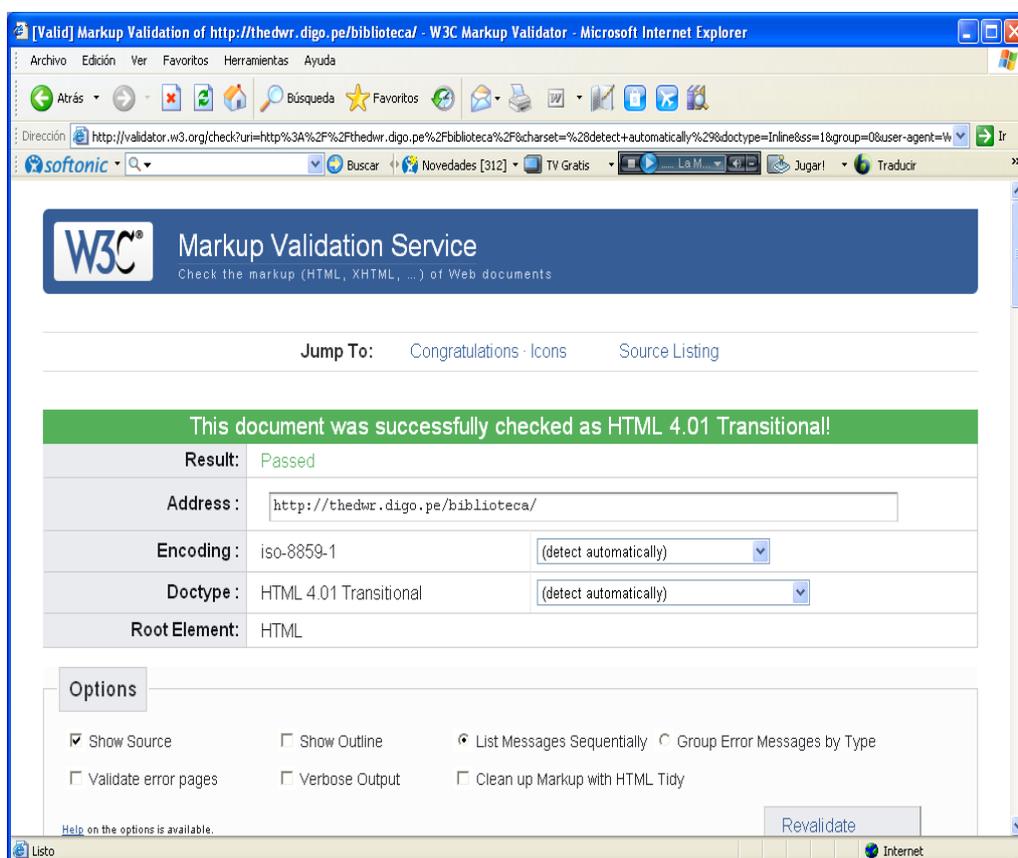
Fuente: Elaboración propia

### 4.3.1 Validación mediante w3c

#### 4.3.1.1 Validación del código fuente

La siguiente ventana muestra la validación obtenida mediante la validación w3c del al código fuente de la página del Software, DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO – 2008

**Figura 16. Validación W3C del código fuente del software**



Fuente: [www.w3c.es](http://www.w3c.es) Consorcio World Wide Web (W3C)

A continuación, se muestra un gráfico del permiso obtenido para mostrar los iconos que indican que el código fuente ha sido validado exitosamente.

**Figura 17. Validación W3C del software**

Congratulations

The document located at <http://thedwr.digo.pe/biblioteca/> was successfully checked as HTML 4.01 Transitional. This means that the resource in question identified itself as "HTML 4.01 Transitional" and that we successfully performed a formal validation using an SGML, HTML5 and/or XML Parser(s) (depending on the markup language used).

*"valid" icon(s) on your Web page*

To show your readers that you have taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the HTML you could use to add this icon to your Web page:

```


<p>
  <a href="http://validator.w3.org/check?uri=referer"></a>
</p>


<p>
  <a href="http://validator.w3.org/check?uri=referer"></a>
</p>
  
```

A [full list](#) of icons, with links to alternate formats and colors, is available: If you like, you can download a copy of the icons to keep in your local web directory, and change the HTML fragment above to reference your local image rather than the one on this server.

Fuente: [www.w3c.es](http://www.w3c.es) Consorcio World Wide Web (W3C)

**4.3.1.2 Validación de las hojas de estilo**

La siguiente ventana muestra la segunda validación obtenida de la W3C con respecto a las hojas de estilo css.

**Figura18. Validación W3C de las hojas de estilo CSS del software**



Fuente: [www.w3c.es](http://www.w3c.es) Consorcio World Wide Web (W3C)

### 4.3.1.2 Métrica puntos de función

#### Calculo de puntos de función sin ajustar:

Función de iteración del usuario del software vía Web de consultas bibliográficas detalladas para la biblioteca especializada de la FINESI – UNA - PUNO – 2008.

**Tabla 9. Parámetros de medición en los módulos complejo y mediano**

Parámetro de medición	Cuenta	Complejo			Mediano		
		Factor de ponderación	Subtotal	Cuenta	Factor de ponderación	Subtotal	Cuenta Total
Número de entradas de usuario	10	3	30	12	3	36	66
Número de salidas de usuario	15	4	60	27	4	108	168
Número de consultas del usuario	17	3	51	16	3	48	99
Número de archivos	19	7	133	17	7	119	252
Número de interfaces externas	2	5	10	0	5	0	10
<b>Total, de PF =</b>							<b>595</b>

Fuente: (Roger S. Pressman, 1998)

**Calculo de los valores de ajuste de la complejidad según las respuestas a las siguientes preguntas:**

**Tabla 10. Indicadores del Valor de ajuste de complejidad.**

Indicador de VAC	Sin influencia	Incidental	Moderado	Medio	Significativo	Esencial
VAC	0	1	2	3	4	5

Fuente: (Roger S. Pressman, 1998)

**Tabla 11. Factores de medición del peso de complejidad**

<b>N</b>	<b>Factor de peso de complejidad</b>	<b>VAC</b>
1	¿Requiere el sistema copias de seguridad y de recuperación fiables?	2
2	¿Se requiere comunicación de datos?	2
3	¿Existen funciones de procesamiento distribuido?	0
4	¿Es crítico el rendimiento?	1
5	¿Se ejecutaría el sistema en un entorno operativo existente y fuertemente utilizado?	3
6	¿Requiere el sistema entrada de datos interactiva?	0
7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	0
8	¿Se actualizan los archivos maestros de forma interactiva?	4
9	¿Son complejos las entradas, las salidas, los archivos o las peticiones?	2
10	¿Es complejo el procesamiento interno?	2
11	¿Se ha diseñado el código para ser reutilizable?	1
12	¿Están incluidas en el diseño la conversión y la instalación?	0
13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	4
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?	4
Total		<b>25</b>

**Fuente:** Elaboración propia

Para calcular los puntos de función (PF), se utiliza la relación siguiente:

$$PF = \text{cuenta-total} \times [ 0,65 + 0,01 \times \Sigma (Fi) ] = 535.5 = 536 \text{ puntos de función}$$

Punto de función por mes : 6 Meses

Punto de función al mes :  $535.5/6 \text{ meses} = 89.25$

Costo por punto de función :  $2000/89.25 = 22.41$

Costo del software: 12000

#### 4.4 Prueba Estadística

Para determinar el aporte de este trabajo se plantea lo siguiente la prueba de proporciones par una sola muestra.

##### PLANTEAMIENTO DE HIPÓTESIS

###### i) Formulación de la hipótesis

a) **Hipótesis Nula:**  $H_0: P \leq 80\%$  La implementación del Software de consulta virtual no mejora significativamente la atención al usuario en la Biblioteca Especializada de la FINESI-UNA - PUNO – 2008.

b) **Hipótesis Alternativa:**  $H_a: P > 80\%$  La implementación del Software de consulta virtual mejora significativamente la atención al usuario en la Biblioteca Especializada de la FINESI-UNA - PUNO – 2008.

ii) Nivel de significancia  $\alpha = 0.1$  el valor de tabla de Z para una prueba de una sola cola es igual a 1.65

iii) Prueba estadística

Esta prueba comprende el cálculo del valor estadístico de prueba Z

$$Z_{prueba} = \frac{\frac{x}{n} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}$$

Donde:

$x$  = *ocurrencias*

$n$  = *observaciones*

$\frac{x}{n}$  = *proporción de la muestra*

$p_0$  = *proporción propuesta*

$$\sqrt{\frac{p_0(1-p_0)}{n}} = \text{desviación estándar de la proporción}$$

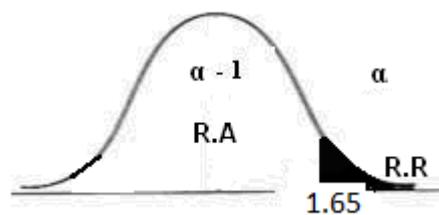
Si se muestrea a partir de una población finita

$$\frac{n}{N} \cdot 100\% > 5\%$$

Se debe utilizar el factor finito de corrección

$$Z_{prueba} = \frac{\frac{x}{n} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n} \cdot \frac{N-n}{N-1}}}$$

iv) Criterios de decisión



v) Cálculo: (Tratamiento de datos)

**Tabla de valoración de índices**

NO MEJORA	SI MEJORA
0	1

**Tabla de resumen de la encuesta con respuesta valorada para ver la aceptación del sistema**

**Tabla 12. Resultados obtenidos**

Sem.	Muestra	Indicador	Total	p	Sem.	Muestra	Indicador	total	p
		Usuario					Usuario		
I	1	1	8	1	VI	1	26	7	1
	2	2	8	1		2	27	8	1
	3	3	7	1		3	28	8	1
	4	4	4	0		4	29	7	1
	5	5	0	0		5	30	7	1
II	1	6	4	0		6	31	8	1
	2	7	8	1	VII	1	32	8	1
	3	8	8	0		2	33	7	1
	4	9	7	1		3	34	6	1
	5	10	8	1		4	35	8	1
III	1	11	1	0	VIII	1	36	8	1
	2	12	8	1		2	37	8	1
	3	13	7	1		3	38	8	1
	4	14	8	1		4	39	8	1
IV	5	15	7	1	IX	1	40	6	1
	1	16	8	1		2	41	8	1
	2	17	8	1		3	42	8	1
	3	18	8	1	X	1	43	8	1
	4	19	7	1		2	44	8	1
5	20	5	1	3		45	8	1	
1	21	8	1	4		46	8	1	
V	2	22	7	1	5	47	8	1	
	3	23	8	1	6	48	8	1	
	4	24	8	1	7	49	8	1	
	5	25	8	1	TOTAL				44

**Fuente:** Elaboración propia

En la tabla 12 se muestra el resumen de calificación que él, usuario otorga al software teniendo como criterio lo siguiente:

**En donde:**

Puntaje que califica el usuario	
0-4	No mejora
5-8	Si mejora

Fuente elaborado por el investigador

Por lo que 44 de ellos opinan que el software si mejora la atención al usuario en la biblioteca especializada de la FINESI-UNA-PUNO.

realizando el cálculo correspondiente se obtiene:

	Z	1.892	Valor de Z calculado
Ocurrencia	X	44	Número de alumnos que consideran el software mejora significativamente la atención al usuario
observaciones	N	48.87	Tamaño de muestra obtenido
Proporción propuesta	Po	0.8	Proporción de alumnos propuestos a los que satisface el sistema
Población	N	344	Población de alumnos matriculados en la FINESI-UNA-PUNO semestre 2008-I.

Como:

$$\frac{n}{N} = \frac{49}{344} \times 100 = 14.24 > \text{a } 5\%$$

entonces usamos el factor finito de corrección y calculamos el valor para Z

$$Z_{\text{Calculado}} = \frac{\frac{44}{49} - 0.8}{\sqrt{\frac{0.8(1-0.8)}{49} \cdot \frac{(344-49)}{(344-1)}}} = 1.891550$$

$$Z_{\text{calculado}} = 1.891550$$

Por consiguiente el valor de  $Z_c > Z_t$  entonces se rechaza la  $H_0$

vi) Decisión  $Z_c > Z_t$  Se rechaza a hipótesis nula en consecuencia se acepta la hipótesis alterna. Se puede concluir con un 90 por ciento de confianza que, la implementación del Software de consulta virtual mejora significativamente la atención al usuario biblioteca especializada en la FINESI-UNA- PUNO-2008.

## CONCLUSIONES

1. Se ha logrado diseñar el Software de consulta virtual utilizando el Modelo espiral y la metodología TOP DOWN logrando de esta manera mejorar el servicio atención al usuario en la Biblioteca Especializada de la FINESI-UNA – PUNO. Corroborando a esta afirmación con un 90% de confianza la prueba estadística de proporciones para una sola muestra.
2. Se ha analizado y diseñado módulos de consultas bibliográficas específicas realizando entrevistas en forma directa y presentación de prototipos a los usuarios esto para definir los requerimientos del usuario e implementados con: Lenguaje de programación Php, Java Script; y tecnología Ajax ( para mejorar el rendimiento y disminuir el tiempo de conexión entre el usuario y el servidor ).
3. Se ha implementado módulos de información adicional permitiendo facilitar al usuario la elección de su consulta
4. Se ha implementado módulos de préstamos y devoluciones de bibliografía que permitieron agilizar estas tareas.
5. Se logró diseñar una interfaz de fácil manipulación para el usuario gracias a la presentación previa de prototipos y utilización de hojas de estilo (CSS) para facilitar el formato de las páginas.

## RECOMENDACIONES

1. Para desarrollar futuros proyectos con tecnología Web se recomienda utilizar tecnología AJAX (Unión de varias tecnologías independientes que se unen de formas nuevas y sorprendentes), que mejoran notablemente la velocidad de conexión entre el usuario y el servidor.
2. Se recomienda que para la codificación de textos existentes en la biblioteca especializada de la FINESI - UNA - PUNO se utilice la codificación de la tabla DEWEY
3. Se recomienda seguir investigando acerca de la forma de integrar todas las bibliotecas especializadas existentes en la UNA – PUNO
4. Se recomienda realizar prototipos de interfaz para la recopilación de información sobre los requerimientos del usuario.

## BIBLIOGRAFÍA

## Área informática

- Coronel, G. (2005). **Creando soluciones web con php 4.5 y 5.x y MySQL Server para versiones 3.2.3, 4.x y 5.x**. Perú: Macro EIRL.
- Crovetto, C. (2005). **Oracle 10g: Base de datos**. Perú: Megabyte. ISBN
- Fernández, E. I. (2004). **E-learning: Implementación de proyectos de formación online**. España: RA-MA
- Guzmán, R. (2001). **Base de datos relacionales: Un enfoque práctico**. Chile: Fondo de Desarrollo.
- Pedregosa M. Á. (2002), **Diseño y Programación de páginas Web**. Argentina: thales SRL.
- Pressman, R. (2007). **Ingeniería del Software un enfoque práctico**. México: McGraw-Hill
- Pressman, R. (1998). **Ingeniería del Software un enfoque práctico**. México: McGraw-Hill.
- Sommerville I. (2005), **Ingeniería del software**. España: Top Printer Plus S.L.L.

**Área estadística:**

Allen, W. (2003). ***Estadística Aplicada a los negocios y la economía.***

Colombia: Mc Graw-Hill.

INEI, ***Guía práctica construcción e interpretación de indicadores***

***estadísticos, regionales.*** Lima: Instituto Nacional de Estadística e

Informática.

Mendoza, F y Mendoza, F. R. (2005) ***Estadística para ciencias de la***

***educación:*** Titikaka

Mitacc, M. (1998). ***Tópicos de estadística descriptiva y probabilidad.*** Perú:

San Marcos

Moya, R. y Saravia A. G. (2000), ***Probabilidad e Inferencia Estadística,*** Perú:

San Marcos

## WEBGRAFIA

Consortio internacional World Wide Web (W3C) foro neutral para la creación de estándares Web. Disponible en <http://www.w3c.es>. Accesado el 20 de diciembre del 2008

José Manuel Pérez desarrollo Web de bases de datos. Disponible en <http://www.espestudio.com/articulo/desarrollo-web/bases-de-datos-mysql/Que-es-MySQL.htm>. Accesado el 16-08-2005

Jesse James Garrett. Ajax: Un Nuevo acercamiento a las Aplicaciones Web. Disponible en <http://www.librosweb.es/ajax/capitulo1.html>. Accesado el 28-05-2008.

Encriptación MD5. Disponible en: [http://php.ciberaula.com/articulo/enciptacion\\_md5\\_php](http://php.ciberaula.com/articulo/enciptacion_md5_php). Accesado el 30-01-2009

[http://es.wikipedia.org/wiki/Base\\_de\\_datos#Bases\\_de\\_datos\\_est.C3.A1ticas](http://es.wikipedia.org/wiki/Base_de_datos#Bases_de_datos_est.C3.A1ticas) .  
Accesado el 30-01-2009

Villalobos, C. (2008). **Manejo de plataformas web Linux, Apache, Mysql, PHP** [en línea]. Texinfo. Edición <http://es.wikipedia.org/w/index.php?title=Bottom-up&redirect=no>.

[http://es.wikipedia.org/wiki/Base\\_de\\_datos#Bases\\_de\\_datos\\_est.C3.A1ticas](http://es.wikipedia.org/wiki/Base_de_datos#Bases_de_datos_est.C3.A1ticas).  
Accesado el 30-01-2009

[http://es.wikipedia.org/wiki/Base\\_de\\_datos](http://es.wikipedia.org/wiki/Base_de_datos). Accesado el 30-01-2009

[http://es.wikipedia.org/wiki/Clave\\_for%C3%A1nea#Segunda\\_Forma\\_Normal\\_282FN.29](http://es.wikipedia.org/wiki/Clave_for%C3%A1nea#Segunda_Forma_Normal_282FN.29). Accesado el 27-02-2009

[http://es.wikipedia.org/wiki/Clave\\_for%C3%83%C2%A1nea#Tercera\\_Forma\\_Normal\\_283FN.29](http://es.wikipedia.org/wiki/Clave_for%C3%83%C2%A1nea#Tercera_Forma_Normal_283FN.29). Accesado el 27-02-2009

<http://es.wikipedia.org/wiki/Internet>. Accesado el 27-02-2009

<http://www.monografias.com/trabajos7/html/html.shtml>. Accesado el 16-04-2008

tANC/mem2002/ ponencia Jaime Sánchez CHILE.doc>; Internet; accesado el 25 de Julio de 2008. Edición. 2.1. [Madrid - España]. CiberAula, Disponible en Web: <[http://www.ciberAula.com/curso/lamp/que\\_es/](http://www.ciberAula.com/curso/lamp/que_es/)>. Internet: accesado el 10 de diciembre del 2008.

# ANEXOS

Anexo 01

¿Qué atributos le gustaría a Ud. Tener en el SOFTWARE DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO - 2008 del que se te muestra ahora?

1er prototipo entregado al usuario final para la obtención de requerimientos que se convirtieron luego plasmados en el software.

N	CODIGO	TITULO	AUTOR	AREA
1	LDP00578	PowerBuilder Desarrollo de Aplicaciones CLIENTE - SERVIDOR	Paul mahler	Informática
2	LDM00102	ECUACIONES DEFERENCIALES Y PROBLEMAS	WILLIAM E. BOYCE	Matemáticas
3	LDLN00241	INTRODUCCION A LA PROGRAMACION LINEAL	GRACIELA BUENO DE A.	Informática
4	LDR00245	REDES NEURONALES #goritmos, Aplicaciones	James A. Freeman	Informática
5	LDP00578	PowerBuilder Desarrollo de Aplicaciones CLIENTE	Paul mahler	Informática
6	LDM00102	ECUACIONES DEFERENCIALES Y PROBLEMAS	WILLIAM E. BOYCE	Matemáticas
7	LDLN00241	INTRODUCCION A LA PROGRAMACION	GRACIELA BUENO DE A.	Informática
8	LDR00245	REDES NEURONALES #goritmos	James A. Freeman	Informática
9	LDP00578	PowerBuilder Desarrollo de Aplicaciones	Paul mahler	Informática
10	LDM00102	ECUACIONES DEFERENCIALES Y PROBLEMAS	WILLIAM E. BOYCE	Matemáticas

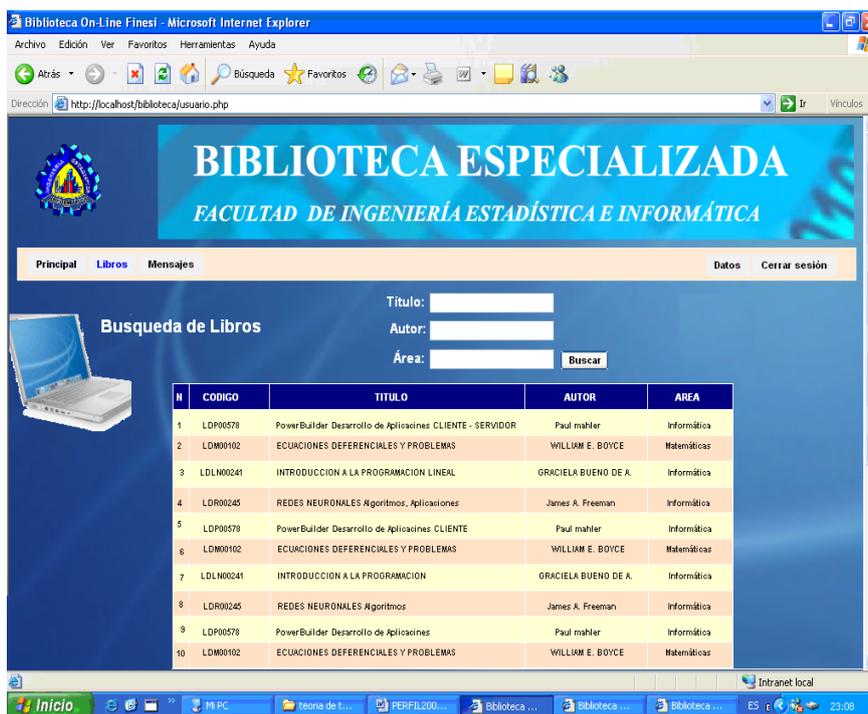
## Anexo 02

**2do prototipo** entregado al usuario final para la obtención de nuevos requerimientos y prueba de la funcionalidad de los anteriores requerimientos obtenidos.

En esta presentación se formula las siguientes preguntas:

**¿Es agradable los colores que se presentan en el interfaz?**

**¿Que otro atributo te agradecería tener en el SOFTWARE VÍA WEB PARA LA ADMINISTRACIÓN DE LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA- PUNO – 2008?**

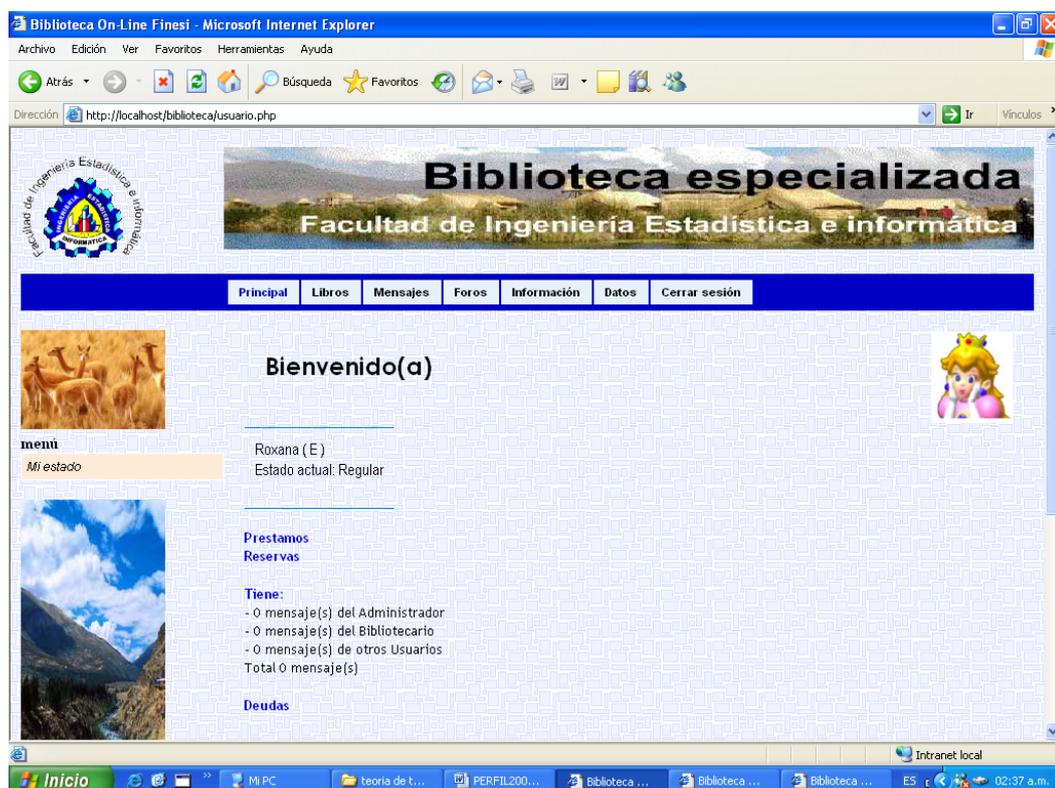


N	CODIGO	TITULO	AUTOR	AREA
1	LDR00578	PowerBuilder Desarrollo de Aplicaciones CLIENTE - SERVIDOR	Paul mahler	Informática
2	LDM00102	ECUACIONES DIFERENCIALES Y PROBLEMAS	WILLIAM E. BOYCE	Matemáticas
3	LDLN00241	INTRODUCCION A LA PROGRAMACION LINEAL	GRACIELA BUENO DE A.	Informática
4	LDR00245	REDES NEURONALES Algoritmos, Aplicaciones	James A. Freeman	Informática
5	LDR00578	PowerBuilder Desarrollo de Aplicaciones CLIENTE	Paul mahler	Informática
6	LDM00102	ECUACIONES DIFERENCIALES Y PROBLEMAS	WILLIAM E. BOYCE	Matemáticas
7	LDLN00241	INTRODUCCION A LA PROGRAMACION	GRACIELA BUENO DE A.	Informática
8	LDR00245	REDES NEURONALES Algoritmos	James A. Freeman	Informática
9	LDR00578	PowerBuilder Desarrollo de Aplicaciones	Paul mahler	Informática
10	LDM00102	ECUACIONES DIFERENCIALES Y PROBLEMAS	WILLIAM E. BOYCE	Matemáticas

### Anexo 03

**3er prototipo** entregado al usuario final para la obtención de más requerimientos y pruebas de aceptación de interfaz.

En el siguiente prototipo es entregado al usuario final con todos los requerimientos obtenidos de los anteriores, para su uso y manipulación real.



Biblioteca On-Line Finesi - Microsoft Internet Explorer

http://localhost/biblioteca/usuario-libros.php?w=4&r=2

### Resultado de busqueda

Existen 58 libros que mostrar.

Código	Título	Autor	Área
11797	Theory and Problems of Statistics	SPIEGEL, Murray R.	Estadística
12575	Cálculo Diferencial e Integral	GRANVILLE, William Anthony; BYNGTON, Steven T., trad.	Matemáticas
1477	Matemáticas Superiores para Ingenieros	REDDICK, H.W.; miller, f.h.; VAZQUEZ BRIONES, Hector, trad.	Matemáticas
14813	Cálculo con Geometría analítica	THOMAS, George B.; FINNEY, Ross L.; LOPEZ MATEOS, Manuel, trad.	Matemáticas
14880	Estadística no Paramétrica, Aplicada a las Ciencias de la Conducta	SIEGEL, Sidney; AGUILAR VILLALOBOS, Javier, trad.	Estadística
15290	Calculus With Analytic GeometryCálculo con Geometría Analítica	SWOKOWSKI, Earl	Matemáticas
15881	Resolución de Problemas de Estadística Aplicada a las Ciencias Sociales: (Guía Práctica para Profesores y Alumnos)	FERNANDEZ DIAZ, María José; GARCIA RAMOS, José Manuel; ASENSIO MUÑOZ, Isabel Inmaculada y otros	Tesis
16220	Probabilidad y Aplicaciones Estadísticas	MEYER, Paul L.; PRADO CAMPOS, Carlos, trad.	Estadística
16310	Cálculo con Geometría Analítica	ZILL, Dennis G.; OJEDA PEÑA, Eduardo, trad.	Matemáticas
18101	Cálculo	SMITH, B.; trad.	

**menú**

- Buscar libro
- Reservar Libros
- Libros más solicitados
- Enviar trabajo

**Poco espacio en disco**  
Se está quedando sin espacio en FAMILIA (F.). Para liberar espacio en esta unidad y borrar los archivos antiguos o innecesarios, haga clic aquí...

Biblioteca On-Line Finesi - Microsoft Internet Explorer

http://localhost/biblioteca/usuario-libros.php?w=4&r=1&wcodigo=TR0001

### Resultado de busqueda

Existen 1 libros que mostrar.

<b>Código</b>	TR0001
<b>Stock</b>	0
<b>Prestados</b>	0
<b>Reservados</b>	0
<b>Libres</b>	0
<b>Revisado</b>	0
<b>Área</b>	Informática
<b>Autor</b>	FLORES CARMONA NICOLÁS JOHNATAN
<b>Observaciones</b>	PARA OBTENER EL TITULO DE INGENIERO EN SISTEMAS COMPUTACIONALES Con la asesoría de: M.C.C. MARTHA ALICIA ROCHA SÁNCHEZ

**REINGENIERIA**

INTRODUCCIÓN  
CAPÍTULO 1: REINGENIERIA DE PROCESOS  
CAPÍTULO 2: SISTEMAS DE INFORMACIÓN HEREDADOS Y REINGENIERIA DE SOFTWARE  
CAPÍTULO 3: METODOS Y MODELOS DE LA REINGENIERIA DE SOFTWARE  
CAPÍTULO 4: RECONSTRUCCIÓN DE LA ARQUITECTURA  
CONCLUSIONES

ver tema

Regresar

**Anexo 04****ENCUESTA 01**

Cuestionario de preguntas (encuesta) dirigida para probarla funcionalidad e interfaz del SOFTWARE DE CONSULTA VIRTUAL PARA LA ATENCIÓN AL USUARIO EN LA BIBLIOTECA ESPECIALIZADA DE LA FINESI-UNA - PUNO - 2008

**I. Permite realizar consultas bibliografías específicas (Título, Autor, Índice y Área)****1.1 ¿Le ayuda el software a realizar consultas mediante Título, ¿Autor, Índice y Área?**

- a) Si Ayuda
- b) Ayuda regular
- c) No Ayuda

**II Muestra información adicional para facilitar la elección de su consulta****2.1 ¿Le ayuda la información de los datos estadístico y sugerencias que encontró en el software para encontrar el tema que buscaba (Estadísticas de libros más consultados, sugerencias en foros)?**

- a) Si Ayuda
- b) Ayuda regular
- c) No Ayuda

**III Permiten agilizar la atención los módulos de préstamos y devoluciones de consultas bibliográficas**

- a) Si Ayuda
- b) Ayuda regular
- c) No Ayuda

**IV Le ayuda la interfaz a manipular con facilidad los módulos del software****1. ¿Le pareció difícil manipular el software?**

- a) Si
- b) Regular
- c) No .

Anexo 05

Tabla Resultados de la encuesta para conocer el nivel de aceptación del software.

Sem	Muestra	Indicador	Permite realizar consultas bibliográficas específicas (Título, autor, índice área)		Muestra información adicional para facilitar la elección de las consultas		Permite realizar la atención al usuario los módulos de préstamos y devolución de bibliografía		Permita la interfaz manipular con facilidad módulos del software		Total
			No ayuda	Regular	Si ayuda	No ayuda	Regular	Si ayuda	No ayuda	Regular	
I	1	Usuario			2	2					8
	2				2	2					8
	3				2	2					7
	4		1		2	2	1	1	1	2	4
	5		0		0	0	0	0	0	0	4
II	6		1		2	2	1	1	1	2	4
	7				2	2					8
	8				2	2					2
	9				2	2	1	1	2	2	7
	10				2	2					8
III	11		1		2	2					1
	12				2	2					8
	13		1		2	2					7
	14				2	2	1	1	2	2	8
	15				2	2					7
IV	16				2	2					8
	17				2	2					8
	18				2	2					8
	19				2	2					7
	20		1		2	2			1	2	5
V	21				2	2					8
	22				2	2					7
	23				2	2			1	2	8
	24				2	2					8
	25				2	2					9
VI	26				2	2			1	2	8
	27				2	2					8
	28				2	2					8
	29				2	2					8
	30				2	2			1	2	7
VII	31				2	2					8
	32				2	2					7
	33				2	2			1	2	8
	34				2	2					8
	35				2	2	1	1		2	6
VIII	36				2	2					8
	37				2	2					8
	38				2	2					8
	39				2	2					8
	40				2	2					6
IX	41				2	2					8
	42				2	2					8
	43				2	2					8
	44				2	2					8
	45				2	2					8
X	46				2	2					8
	47				2	2					8
	48				2	2					8
	49				2	2					8

Fuente: Elaborado por el investigador

