



*INSTITUTO SUPERIOR MINERO METALÚRGICO*

*“Dr. Antonio Núñez Jiménez”*

*Facultad: Metalurgia – Electromecánica*

*Carrera de Ingeniería Informática*

# *Trabajo de Diploma*

*Para optar por el título de Ingeniero  
Informático*

**Automatización de procesos penales en la Fiscalía  
Provincial Holguín**

**Autor:** Javier Calzadilla López

**Tutor:** Ing. Junier García Nicot

**Moa, Cuba**

**Julio, 2011**

# *Pensamiento.*

*Pensar es el trabajo más difícil que existe. Quizá sea ésta la razón por la que haya pocas personas que lo practiquen.*

*Henry Ford.*

# *Agradecimiento.*

*Quiero agradecer a todas las personas que de una forma u otra me han servido de apoyo para el desarrollo de esta investigación y en especial:*

*A mi familia que estuvo conmigo en cada momento.*

*A mi tutor por su apoyo durante el desarrollo de este trabajo.*

*A todos los profesores que a lo largo de estos cinco años dieron lo mejor de si para conmigo.*

*A mis compañeros de clases que más que eso hoy son mis amigos.*

*A todos y a cada uno de ustedes*

*Mi eterno agradecimiento.*

## *Dedicatoria.*

*Dedico esta investigación de forma muy especial:*

*A mis padres que siempre han sido el pilar que sustenta mi vida.*

*A mi hermanita por ser la musa que me inspira.*

*A mis tíos por su apoyo incondicional, sus consejos y sus enseñanzas.*

*A todo el resto de mi familia que aunque no están plasmada en este papel, están  
presente en mi corazón.*

*A mi novia por su amor, belleza, comprensión y ser lo más especial que busca todo  
hombre pero que yo ya encontré.*

**DECLARACION DE AUTORIA.**

Declaro que soy el único autor de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa y a la Fiscalía Provincial de Holguín para que hagan uso de este trabajo según se estime conveniente.

Para que así conste firmo la presente a los\_\_\_\_ días del mes\_\_\_\_\_ del año 2011.

\_\_\_\_\_  
Firma del autor

\_\_\_\_\_  
Firma del tutor

## **RESUMEN.**

El ascendente desarrollo que ha acontecido en los últimos años ha situado las tecnologías de la información en un lugar relevante en el ámbito del intercambio y gestión de la información e informatización de los diferentes sistemas.

La implantación en la sociedad de las denominadas Tecnología de la Informática y las Comunicaciones (TIC), produce cambios insospechados en todas las facetas de la vida. Cada vez son más las organizaciones e instituciones que se inclinan por incorporar aplicaciones que gestionen su información, obteniendo así una mayor dinámica en sus procesos.

La Fiscalía Provincial de Holguín se desenvuelve en un entorno de niveles y volúmenes considerables de información provenientes de todos los municipios de dicha provincia, esta información es manejada manualmente. La gestión de la información se ha convertido en un tema de amplia significación, lo que propició la creación de este sistema automatizado que permite llevar el control y manejo de la información de los expedientes en los procesos penales relacionados con lo menores víctimas.

Con el desarrollo de esta investigación se logró la informatización de la gestión de la información de los procesos penales llevados a cabo por dicha entidad a través de la creación de una aplicación Web que gestiona la información de manera ágil, segura y eficiente.

Para la elaboración de la misma se realizó un estudio bibliográfico sobre las aplicaciones Web y las herramientas para la construcción de las mismas. En este documento se recoge un resumen del estudio realizado, se presenta la metodología de Ingeniería de Software que se siguió para el desarrollo del software que se propone como solución de la problemática encontrada.

## **ABSTRACT.**

Highest development been happened in last years has placed at significant site the information technologies relating to information agency interchange and computing of different systems.

The implantation in the society of named TIC, make change unsuspected in all the facets of life. Each time organizations are further and institutions that applications fall for incorporating themselves that they try to obtain his information, obtaining a bigger dynamics in his processes thus.

The Provincial Office of Holguín is unwrapped in an environment of levels and considerable volumes of information coming from all the municipalities of this county, this information is managed manually. The administration of the information has become a topic of wide significance, what propitiated the creation of this automated system that allows to take the control and handling of the information of the files in the penal processes related with the smallest victims.

With the development of this investigation the informatization of the administration of the information of the penal processes was achieved carried out by this entity through the creation of an application Web that negotiates the information in an agile, sure and efficient way.

For the elaboration of the same one he was carried out a bibliographical study on the Web applications and the tools for the construction of the same ones. In this document a summary of the realized study is picked up, the methodology of Engineering of Software is presented that was continued for the development of the software that intends as solution of the opposing problem.

## ÍNDICE.

<b>INTRODUCCIÓN</b> .....	1
<b>CAPÍTULO I FUNDAMENTACIÓN TEÓRICA</b> .....	6
1.1 Introducción .....	6
1.2 Estado del arte .....	6
1.3 Tendencias y tecnologías actuales .....	7
1.3.1 Tecnología Cliente - Servidor .....	7
1.3.2 Lenguaje de programación .....	8
1.3.3 Justificación del lenguaje de programación a utilizar .....	11
1.3.4 Sistemas Gestores de Base de Datos (SGBD) .....	12
1.3.5 Justificación del Sistema Gestor de Base de Datos a utilizar .....	14
1.3.6 Tecnología RIA (Rich Internet Application) .....	15
1.3.7 Tecnologías y herramientas de trabajo .....	16
1.4 Metodologías ágiles .....	18
1.4.1 Scrum .....	22
1.4.2 XP (Extreme Programming) .....	23
1.4.3 Valores de la metodología XP .....	23
1.4.4 Prácticas de la metodología XP .....	25
1.4.5 Fases de la metodología XP .....	29
1.5 Conclusiones .....	31
<b>CAPÍTULO 2: ANÁLISIS Y DISEÑO</b> .....	32
2.1 Introducción .....	32
2.2 Planificación – Definición .....	32
2.2.1 Concepción del sistema .....	32
2.2.2 Modelo de Historias de usuario del negocio .....	34
2.2.3 Lista de reserva del producto (LRP) .....	35
2.2.4 Historias de usuario .....	37
2.2.5 Valbración de riesgos .....	38
2.2.6 Modelo de Diseño .....	40
2.3 Conclusiones del capítulo .....	40
<b>CAPÍTULO 3: DESARROLLO DEL SISTEMA</b> .....	41
3.1 Introducción .....	41

3.2 Desarrollo .....	41
3.2.1 Plan de releases.....	41
3.2.2 Cronograma de Producción .....	42
3.2.3 Tarjetas CRC .....	44
3.2.4 Tareas por Historias de Usuario.....	44
3.3 Conclusiones del capítulo .....	45
<b>CAPÍTULO 4: PRUEBAS.....</b>	<b>46</b>
4.1 Introducción.....	46
4.2 Pruebas .....	46
4.2.1 Pruebas de aceptación .....	46
4.3 Conclusiones del capítulo.....	47
<b>CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD.....</b>	<b>48</b>
5.1 Introducción.....	48
5.2 Efectos Económicos .....	48
5.2.2 Efectos indirectos .....	49
5.2.3 Efectos externos.....	49
5.2.4 Intangibles.....	49
5.3 Beneficios y Costos Intangibles en el Proyecto .....	50
5.3 Ficha de costo .....	50
5.4 Conclusiones del capítulo.....	54
<b>CONCLUSIONES GENERALES .....</b>	<b>54</b>
<b>RECOMENDACIONES.....</b>	<b>56</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>57</b>
<b>BIBLIOGRAFÍAS.....</b>	<b>58</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>60</b>

## **INTRODUCCIÓN.**

Desde los orígenes de la humanidad el hombre en su afán por la supervivencia se ha visto inmerso en la necesidad de recurrir a la toma de decisiones con el fin de adaptarse a las diferentes condiciones en que se ha desarrollado.

La situación económica y social que caracteriza a la sociedad moderna genera profundos cambios en las organizaciones, las cuales se preparan para ser más flexibles y establecen estrategias con el objetivo de adaptarse al entorno altamente turbulento en el que desarrollan sus acciones. Ante ambientes tan poco estables y la imposibilidad de actuar a ciegas, los miembros de las organizaciones y, en particular, sus altas gerencias necesitan manipular grandes volúmenes de información para cumplir con sus funciones esenciales. Deben implementarse entonces, prácticas administrativas dirigidas a garantizar el éxito organizacional y entre ellas, la toma de decisiones, soportada en el análisis de información, para lo que ha venido desarrollando mecanismos que avalen o justifiquen dichas decisiones.

La segunda mitad del siglo XX fue testigo de algunos hechos concretos que propiciaron grandes cambios, entre ellos la aparición de las computadoras, el crecimiento acelerado de su velocidad de procesamiento y capacidad para el almacenamiento de datos, además de la facilidad de interconexión, sin dudas, ha posibilitado disponer de servicios de acceso en línea a bases de datos y ha provocado una gran explosión de información que rebasa la capacidad de procesamiento de las organizaciones y la búsqueda de herramientas para el manejo de estos grandes volúmenes de información.

Todo esto, unido al incremento de la competencia a nivel mundial, impulsado por el dominio de las transnacionales, ha originado desde los años ochenta, un contexto de creciente necesidad de dotarse de técnicas de captación y análisis de información sobre el entorno competitivo, tecnológico y, en particular, de formas organizativas y herramientas que faciliten dicho objetivo.

En el mundo actual la mayoría de las actividades, acciones y operaciones hacen uso de la computadora. La informática soporta cambios continuos que repercuten a diario

*Automatización de Procesos Penales.*  
*Javier Calzadilla López.*

---

en nuestra sociedad. Hoy no se conciben entidades que no utilicen las Tecnología de la Información y las Comunicaciones (TIC) ya que esta es capaz de aportar soluciones eficientes y eficaces permitiendo actuar con rapidez, obtener el máximo rendimiento del personal y tomar mejores decisiones.

Nuestro país no se encuentra ajeno al desarrollo del producto software y en los últimos años ha tomado cierto auge con el objetivo siempre de brindar una mejor información a la sociedad. Las universidades cubanas juegan un papel fundamental en este proceso sometido a cambios y transformaciones ante los retos que le plantea la actual sociedad del conocimiento, revelándose cada vez más como componente estratégico en la construcción de una sociedad por el rol social que juegan en la producción y transferencia de conocimientos. La importancia del vínculo entre las universidades y las entidades de producción en general, forma parte del discurso de la sociedad de nuestros días.

Uno de los factores más importantes en el desarrollo del mundo contemporáneo son los conocimientos que posea una organización, cualquiera que esta sea y claro está, como sea capaz de aplicarlos. En este empeño, comprobada está, la marcada importancia del aprovechamiento del imponente caudal de recursos de información existentes para la toma de decisiones en las instituciones. El sólido crecimiento y desarrollo de cualquier economía, desde la de una pequeña organización hasta la de un país entero, dependerá indudablemente de decisiones basadas en el conocimiento.

Apoyándonos en lo antes mencionado y en el siguiente pensamiento: "El acceso rápido y eficiente a una información confiable y precisa permite adoptar una posición adecuada a la hora de tomar una decisión para solucionar un problema con un menor costo. Pero esto sólo es posible, si se ha realizado previamente un proceso de análisis de la información, en el que se adicione un conjunto de valores pertinentes a partir del trabajo intenso que realizan especialistas entrenados en el uso de las técnicas de información".

Por todo lo planteado anteriormente la Fiscalía Provincial de Holguín como órgano guardián de los intereses del Estado cubano lleva a cabo una serie de procesos y controles penales los cuales por su complejidad, seriedad y seguridad sería

conveniente automatizar logrando mejoras en el flujo, manejo y control de la información.

Uno de estos procesos son los relacionados con los menores víctimas donde nuestro país ya desde hace mucho venía desplegando un serio trabajo en función de la protección de la infancia y de garantizar el desarrollo armónico y feliz de su personalidad poniendo de manifiesto lo planteado por nuestro Héroe Nacional José Martí cuando este expresó. “Hay que dar al niño hombros para que sustente el peso que la vida le eche encima, no peso ajeno que oprima sus hombros”.

La Constitución de la República de Cuba de 1976 en su artículo 40 establece que la niñez y la juventud disfrutan de particular protección por parte del Estado y la sociedad y que la familia, la escuela, los órganos estatales y las organizaciones de masas y sociales tienen el deber de prestar especial atención a su formación integral. Especial preocupación ha existido también en nuestro país por la protección jurídica-penal al menor como víctima de actos sexuales corruptores o deshonestos, desde el Código Penal Español de 1870, pasando por el Código de Defensa Social hasta la Ley 1279 de 1973 del poder revolucionario y que mantuvo la Ley No. 21 de 1979 y la Ley No. 62 de 1988 (Códigos Penales).

Debido a la complejidad de este proceso surge la siguiente situación problemática. La entidad antes mencionada es la encargada de llevar un estricto control de procesos penales ya mencionados velando que se cumplan una serie de requisitos, regulaciones y leyes recogidos todos en el Código Penal así como en la Constitución de la República. Para esto es necesario contar con un sin número de informaciones procedentes de todos los municipios de la provincia Holguín lo cual la búsqueda de estos datos en cierto momento resulta complicada debido al volumen de información que crece continuamente retrasando esto en gran medida el desarrollo del proceso. Pese a la importancia de la actividad esta no se realiza de forma automatizada provocando serias dificultades durante el proceso de gestión de la información motivo que conllevó a desarrollar un sistema informático para controlar los datos correspondientes con un proceso penal dado.

Dada esta situación podemos definir como **problema científico**: ¿cómo automatizar la gestión de la información para favorecer el tratamiento, control y seguridad de los datos en la Fiscalía Provincial Holguín?

Por lo antes planteado se tiene como **objeto de estudio** los procesos de desarrollo de los sistemas de gestión de la información.

Paralelo al objetivo que persigue la presente investigación, el **campo de acción** sería la construcción de un sistema de gestión de la información para el seguimiento del curso de los expedientes en la Fiscalía Provincial Holguín.

De manera que el **objetivo general** de la investigación viene encaminado en desarrollar una aplicación Web que permita la gestión y control de la información de forma más rápida, segura y eficiente en la Fiscalía Provincial Holguín.

Para guiar nuestra investigación nos planteamos como **idea a defender** de la misma que si utilizamos las metodologías de desarrollo de software, las herramientas y tecnologías actuales para la construcción del sistema de gestión de información, favoreceríamos con celeridad y calidad el tratamiento, control y seguridad de la información en la entidad en cuestión.

Los **objetivos específicos** a plantearse son los siguientes:

- Realizar un análisis del estado actual de los procedimientos penales.
- Realizar un estudio acerca de las tecnologías y herramientas actuales necesarias para la posterior implementación del sistema.
- Realizar el análisis, diseño e implementación del sistema.
- Garantizar la seguridad de los datos que se gestionan.

Para lograr un mejor desarrollo de la investigación y darle seguimiento al objetivo trazado se plantearon las siguientes **tareas**.

- Estudio del proceso de gestión y control de los procesos penales.
- Análisis de las bibliografías que permitan la familiarización con las principales tecnologías y herramientas utilizadas actualmente para el desarrollo de aplicaciones informáticas.

- Seguimiento de cada una de las etapas de la ingeniería del software hasta la implementación de un sistema que permita el tratamiento de la información en los procesos penales.
- Realización del estudio de la Factibilidad y Sostenibilidad.
- Prueba y Documentación del Sistema.
- Desarrollo del Manual de Usuarios.

Para el cumplimiento de las tareas los métodos de investigación empleados fueron:

**Métodos Teóricos y Métodos Empíricos.**

Los siguientes **Métodos Teóricos** sustentan la investigación:

**Analítico-Sintético:** Permite integrar y descomponer el conocimiento, pues se hizo una breve investigación y estudio del sistema, determinando los aspectos esenciales y el arribo a conclusiones prácticas y teóricas y así identificar el problema concreto existente.

**Inductivo-deductivo:** Permite pasar de lo particular a lo general y viceversa favoreciendo objetivamente el enlace que se establece en la realidad entre lo singular y lo general ya que ambas se complementan mutuamente en el proceso de desarrollo científico.

**Modelación:** Pues se crean abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados.

Los **Métodos Empíricos** empleados son:

**Entrevistas:** Nos permite obtener la información adecuada para adentrarnos más en el problema y en las necesidades de la entidad y así determinar los principales requerimientos del sistema.

**Observación:** Para ver la funcionalidad que existe en la Fiscalía Provincial Holguín y el comportamiento del sistema.

## **CAPÍTULO I FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En este capítulo se realiza un proceso investigativo de los aspectos teóricos necesarios para la elaboración y concepción del Trabajo de Diploma. Se describen los principales conceptos asociados al problema y que son necesarios para un mejor entendimiento y darle solución al mismo. Además se realiza un estudio del proceso de gestión de los procesos penales en Cuba relacionado con la atención y protección al menor víctima estableciendo la estrategia a llevar a cabo y los beneficios esperados. Se hace una caracterización de cada tipo de herramientas y del lenguaje de programación, así como la metodología de desarrollo escogida para el desarrollo del software.

### **1.2 Estado del arte**

En correspondencia con lo postulado en la Convención Internacional sobre los derechos del niño de la que Cuba es signataria, y por la complejidad y características especiales de los asuntos en materia de familia y la experiencia alcanzada al respecto por algunos países, evidencian la conveniencia de que, en los asuntos que se requiera, nuestros órganos veladores de la legalidad, que en este caso son rectorados por la Fiscalía General de la Republica en sus diferentes niveles, puedan recabar la intervención de un equipo técnico asesor multidisciplinario que posibilite minimizar las consecuencias de los delitos donde los menores han resultado víctimas, asesorándolos legalmente y orientándolos en la forma de proceder.

Los avances logrados en el ámbito educativo y cultural de la población cubana requieren de mayores y mejores controles sobre los niños que han sido víctimas de hechos delictivos, en aras de satisfacer hondos intereses humanos, afectivos y sociales de las personas involucradas en dichos procesos.

### **1.3 Tendencias y tecnologías actuales**

Informatización equivale a cambios de mentalidad y de hábitos para asimilar con provecho e incluso con placer el cambio, la sustitución de lo viejo por lo nuevo para que la creatividad se multiplique haciendo el mayor y mejor uso de las muy diversas posibilidades que ofrece “la máquina”, la tecnología. La asimilación de las nuevas tecnologías informáticas, como base para el logro de la sociedad de la información, se convierte en tarea de primer orden. No se trata de estar a la moda o con el último grito de la tecnología, sino de una necesidad inminente de incorporar, en este caso, el paradigma Cliente - Servidor al mayor número de procesos posibles, es evidente que se trata del próximo escalón a subir.

#### **1.3.1 Tecnología Cliente - Servidor**

La arquitectura cliente servidor se basa en la distribución de tareas que desde los años 70 se estableció entre un banco central y sus sucursales. El cliente (un usuario de PC) solicita un servicio (como por ejemplo imprimir) que un servidor le proporciona (un procesador conectado a la LAN). Este enfoque común de la estructura de los sistemas informáticos se traduce en una separación de las funciones que anteriormente formaban un todo. Los detalles de la realización van desde los planteamientos sencillos hasta la posibilidad real de manejar todos los ordenadores de modo uniforme. Para establecer un canal de comunicación entre dos programas que se ejecutan en dos computadoras distintas o incluso en una misma computadora, un programa debe iniciar la conexión y el otro aceptarla. En esto se basa el modelo Cliente-Servidor. El servidor indica al sistema operativo que está en condiciones de aceptar solicitudes de conexión al mismo tiempo que espera por ellas. La mayoría de los servidores atienden solicitudes de varios clientes. Cuando un cliente o computadora remota necesita enviar o descargar información hacia o desde el servidor, éste establece una conexión por donde circula la información y luego la cierra. El cliente interactúa directamente con el usuario, procesa sus solicitudes y muestra los resultados. Así por ejemplo la World

Wide Web utiliza un modelo cliente-servidor, los visualizadores o navegadores (clientes) tienen la función de manipular las solicitudes de documentos hechas por los usuarios. Ellos determinan con qué computadora se realizará la conexión, descargan los documentos solicitados y se lo muestran al usuario en su pantalla. Los servidores Web son los responsables de la otra parte de la conexión, esperan una solicitud proveniente de un usuario y una vez hecha ésta, transmiten el documento al visualizador. La conexión entre el visualizador y el servidor dura exactamente el tiempo justo para que el primero envíe una solicitud y el segundo una respuesta. Esto quiere decir que no existe ninguna conexión entre estos elementos una vez que el usuario ya está leyendo el documento solicitado. Ese modo de trabajo los hace extremadamente eficientes y descongestiona el tráfico en la red [4].

### **1.3.2 Lenguaje de programación**

Un lenguaje de programación es un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina [1].

## **LENGUAJES DEL LADO DEL SERVIDOR**

### **PHP**

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando distintas bibliotecas. El Pre-procesador de hipertextos (Hypertext Pre-processor, PHP por sus siglas en inglés) inicialmente se llamó PHP Tools, siendo publicada bajo licencia de software libre. Sus principales características son:

- Es un lenguaje multiplataforma.

- Soporte para una gran cantidad de bases de datos.
- Integración con varias bibliotecas externas, permite generar documentos en PDF.
- Ofrece una solución simple y universal para las paginaciones dinámicas Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Producto de código abierto.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones.

## **ASP**

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft). Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. El tipo de servidores que emplean este lenguaje son, evidentemente, todos aquellos que funcionan con sistema Windows NT, aunque también se puede utilizar en un PC con Windows 98 si instalamos un servidor denominado Personal Web Server. Incluso en sistemas Linux podemos utilizar las ASP si instalamos un componente denominado Chilisoft, aunque lo ideal es trabajar sobre el servidor Web para el que está pensado: IIS. Con las ASP podemos realizar muchos tipos de aplicaciones distintas. Nos permite acceso a bases de datos, al sistema de archivos del servidor y en general a todos los recursos que tenga el propio servidor. También existe la posibilidad de comprar componentes ActiveX

fabricados por distintas empresas de desarrollo de software que sirven para realizar múltiples usos, como el envío de correo y generar gráficas dinámicamente [4].

## **PERL**

Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el Web. Perl es un acrónimo de Practical Extracting and Reporting Language, que viene a indicar que se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros. Es un lenguaje libre de uso, eso quiere decir que es gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Perl es un lenguaje de programación interpretado, al igual que muchos otros lenguajes de Internet como JavaScript o ASP.

## **JSP**

JSP es un acrónimo de Java Server Pages, que en castellano vendría a decir algo como Páginas de Servidor Java. Es pues, una tecnología orientada a crear páginas Web con programación en Java. Con JSP podemos crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP podremos escribirlas con nuestro editor HTML/XML habitual.

## **Lenguajes del lado del cliente**

### **Java Script**

Java script es un lenguaje de programación interpretado que permite a los desarrolladores crear acciones en sus páginas Web. Es utilizado para crear pequeños programas que luego son insertados en una página Web y en programas más grandes, orientados a objetos pero mucho más complejos.

## **HTML**

El Lenguaje de Marcas de Hipertexto (HyperText Markup Language, HTML por sus siglas en inglés) es el lenguaje de marcado predominante para la construcción de páginas Web. Usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. PHTML es una extensión para un tipo de páginas Web que llevan código PHP y HTML para ser generadas.

### **1.3.3 Justificación del lenguaje de programación a utilizar**

Luego de realizar un estudio de los diferentes lenguajes de programación planteados anteriormente se decide optar PHP del lado del servidor y HTML del lado del cliente.

#### **¿Por qué utilizar PHP?**

Es el acrónimo de Hypertext Pre-Preprocessor. Es software libre, lo que implica menores costes y servidores más baratos que otras alternativas, a la vez que el tiempo entre el hallazgo de un fallo y su resolución es más corto. Es muy rápido. Su sintaxis está inspirada en C. Su librería estándar es realmente amplia, lo que permite reducir los llamados “costes ocultos”, uno de los principales defectos de ASP. PHP se puede considerar multiplataforma. Funciona en toda máquina que sea capaz de compilar su código, entre ellas diversos sistemas operativos para PC y diversos Unix. El código escrito en PHP en cualquier plataforma funciona exactamente igual en cualquier otra. El acceso a las bases de datos de PHP es muy heterogéneo, pues dispone de un juego de funciones distinto por cada gestor. PHP es suficientemente versátil y potente como para hacer tantas aplicaciones grandes que necesiten acceder a recursos a bajo nivel del sistema como pequeños scripts que envíen por correo electrónico un formulario rellenado por el usuario.

### **1.3.4 Sistemas Gestores de Base de Datos (SGBD)**

Los sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos y DataBase Management System. Actualmente existen muchos sistemas gestores de bases de datos, entre ellos, están: MySQL, PostgreSQL y Microsoft SQL Server.

#### **PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, dentro de sus características destacan las siguientes:

##### **Alta concurrencia**

Mediante un sistema denominado MVCC (Acceso concurrente multiversión, por sus siglas en inglés) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo cambios. Esta estrategia es superior al uso de bloqueos por tabla o por filas común en otras bases de datos, eliminando la necesidad del uso de bloqueos explícitos.

##### **Amplia variedad de tipos nativos**

PostgreSQL suministra nativamente soporte para:

- Números de precisión arbitraria.
- Texto de largo ilimitado.
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6).
- Arrays.

Como características adicionales tiene:

Claves ajenas también denominadas Llaves ajenas o **Claves Foráneas**

**Disparadores:** Un disparador se define en una acción específica basada en algo ocurrente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica.

Todos los disparadores se definen por seis características:

- El nombre del disparador.
- El momento en que el disparador debe arrancar.
- La tabla donde el disparador se activara.
- La frecuencia de la ejecución.
- La función que podría ser llamada.

PostgreSQL posee integración con un gran número de lenguajes tales como PHP, Java, C, C++ [1].

## **MySQL**

Es la base de datos de código abierto más popular, esta se encuentra entre las mejores 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.

Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. El modelo relacional se caracteriza a muy grandes rasgos por disponer que toda la información debe estar contenida en tablas, y las relaciones entre datos deben ser representadas explícitamente en esos mismos datos. Es un software de código abierto esto quiere decir que es accesible para cualquiera, para usarlo o modificarlo. MySQL es muy rápido, confiable, robusto y fácil de usar tanto para volúmenes de datos grandes como pequeños. Además tiene un conjunto muy práctico de características desarrolladas en cooperación muy cercana con los

usuarios. Sin embargo bajo constante desarrollo, MySQL hoy en día ofrece un rico y muy útil conjunto de funciones.

La conectividad, velocidad y seguridad hace de MySQL altamente conveniente para acceder a bases de datos [4].

### **SQL Server**

Microsoft SQL Server, propietario de Microsoft, pertenece a la familia de los sistemas de administración de base de datos, operando en una arquitectura cliente/servidor de gran rendimiento. Su desarrollo fue orientado para hacer posible manejar grandes volúmenes de información, y un elevado número de transacciones. SQL Server es una aplicación completa que realiza toda la gestión relacionada con los datos. El servidor sólo tiene que enviarle una cadena de caracteres (la sentencia SQL) y esperar a que le devuelvan los datos. SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones SQL que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL, se trata de procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente.

SQL Server puede manejar perfectamente bases de datos de TeraBytes con millones de registros y funciona sin problemas con miles de conexiones simultáneas a los datos, sólo depende de la potencia del hardware del equipo en el que esté instalado y solamente corre sobre Windows NT- 2000 Server.

### **1.3.5 Justificación del Sistema Gestor de Base de Datos a utilizar**

Luego de realizar un estudio de las diferentes características de los gestores de base de datos expuesto anteriormente se decide optar por MySQL por las siguientes razones:

MySQL es una de las bases de datos más populares desarrolladas bajo la filosofía de código abierto.

Tiene licencia GPL a partir de la versión 3.23.19. Algunas de sus ventajas respecto a otros gestores de base de datos son:

Mayor rendimiento. Mayor velocidad tanto al conectarse con el servidor como al servir selects y demás. Aunque se cuelgue, no suele perder información ni corrompe los datos. Mejor integración con PHP. No tiene límites en el tamaño de los registros. Mejor control y acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos [4].

### **Principales Características**

- El principal objetivo de MySQL es velocidad y robustez.
- Escrito en C y C++.
- Clientes C, C++, JAVA, Perl, TCL.
- Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- Puede trabajar en distintas plataformas y Sistemas Operativos distintos.
- Sistema de contraseñas y privilegios muy flexibles y seguros.
- Utilidad (Isamchk) para chequear, optimizar y reparar tablas.
- Todos los datos están grabados en formato ISO8859\_1.
- Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- El servidor soporta mensajes de error en distintas lenguas.
- Todos los comandos tienen -help o - Para las ayudas.

### **1.3.6 Tecnología RIA (Rich Internet Application)**

#### **AJAX**

AJAX Acrónimo de Asynchronous JavaScript And XML (JavaScript y XML asíncronos, donde XML es un acrónimo de eXtensible Markup Language), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página

sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma.

Algunas tecnologías por la que está compuesto 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.

Sus **características** principales son:

- Mayor rapidez e interactividad, al estilo aplicaciones de escritorio.
- Reduce significativamente la carga de información continua del servidor, actualizando solamente porciones de la página.

### **1.3.7 Tecnologías y herramientas de trabajo**

Zend Studio o Zend Development Environment

Es un completo entorno de desarrollo integrado para el lenguaje de programación PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS y GNU/Linux. Junto con su contraparte Zend Platform, son la propuesta de Zend Technologies para el desarrollo de aplicaciones Web utilizando PHP, actuando Zend Studio como la parte cliente y Zend Platform como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre.

**Características:**

- No requiere la instalación previa de PHP ni del entorno de ejecución de Java.
- Soporte para PHP 4 y PHP 5.
- Resaltado de sintaxis, auto completado de código, ayuda de código y lista de parámetros de funciones y métodos de clase.

- Plegado de código (comentarios, bloques de phpDoc, cuerpo de funciones y métodos e implementación de clases).
- Inserción automática de paréntesis y corchetes de cierre.
- Sangrado automático y otras ayudas de formato de código.
- Detección de errores de sintaxis en tiempo real.
- Funciones de depuración: Botón de ejecución y traza, marcadores, puntos de parada (breakpoints), seguimiento de variables y mensajes de error del intérprete de PHP. Permite también la depuración en servidores remotos (requiere Zend Platform).
- Instalación de barras de herramientas para Internet Explorer y Mozilla Firefox.
- Manual de PHP integrado.

Zend Studio fue diseñado para usarse con el lenguaje PHP, sin embargo ofrece soporte básico para otros lenguajes Web, como HTML, Javascript y XML.

### **Case Studio**

Es una herramienta profesional con la que podrás diseñar tus propias bases de datos, facilitándote herramientas para la creación de diagramas de relación, modelado de datos y gestión de estructuras (Es compatible con ambos SGBD, MySQL y PostgreSQL).

### **Nusphere o PHPEd**

Es un editor para programadores con soporte para múltiples formatos, similar a otras aplicaciones como PHP Coder. PHPEd facilita tu trabajo de programación con numerosas características de gran utilidad entre las que destacan:

- Completo sistema de ayuda
- Planillas de documento y de fragmentos de código frecuentes
- Código de colores para comandos en PHP, Perl, Javascript, SQL, HTML y más.

Además, esta herramienta incluye un cliente de FTP y un servidor Web integrados, totalmente configurables según tus necesidades de trabajo.

## **Adobe Dreamweaver**

Es una aplicación en forma de estudio (basada en la forma de estudio de Adobe Flash) enfocada en la construcción y edición de sitios y aplicaciones Web basadas en estándares. Creado inicialmente por Macromedia (actualmente producido por Adobe Systems). Es el programa de este tipo más utilizado en el sector del diseño y la programación Web, por sus funcionalidades, su integración con otras herramientas como Adobe Flash y, recientemente, por su soporte de los estándares del World Wide Web Consortium.

## **Servidor Apache**

Apache, es un servidor de protocolo para la transferencia de hipertextos (Hypertext Transfer Protocol, HTTP por sus siglas en inglés) de software libre para plataformas Unix, Windows, y Macintosh, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Las características que lo definen son las siguientes:

- Multiplataforma
- Es un servidor de Web conforme al protocolo HTTP/1.1
- Modular: Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hebras en la versión 2.0.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.

## **1.4 Metodologías ágiles**

Actualmente para desarrollar un proyecto con éxito, debe estar regido por metodología de desarrollo, la cual puede seguir uno o varios modelos de ciclo de vida, o sea, el ciclo

de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto. Una metodología de desarrollo de software es un conjunto de técnicas, herramientas, procedimientos y soporte documental que permite a los desarrolladores definir los elementos necesarios para la construcción de un nuevo producto de software. Es la que durante el proceso de desarrollo del software define “quién está haciendo qué, cuándo y cómo para alcanzar un determinado objetivo.” Mediante la metodología de desarrollo de software se van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener. Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Las Metodologías Ágiles se basan en los siguientes principios:

- Realizar entregas cortas en el tiempo y continuas.
- Dar la bienvenida a los cambios.
- Entregas periódicas y frecuentes que funcionen.
- La comunicación directa es el método más eficiente y efectivo para comunicar información. Intenta evitar el teléfono, correos electrónicos, fax, etc.
- La medida principal de progreso es el software que funciona.
- Buen diseño y calidad técnica.
- La simplicidad es algo básico.

Actualmente existen varias metodologías de desarrollo de software, las cuales se deben estudiar con detenimiento para definir cuál es la más adecuada a usar en el software a construir.

Entre las más conocidas están:

- **Metodología Crystal:** Conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Desarrolladas por Alistair Cockburn. El desarrollo de software se considera un juego cooperativo de invención y comunicación.

- **Dynamic Systems Development Method (DSDM):** DSDM empezó en Gran Bretaña en 1994 como un consorcio de compañías del Reino Unido que querían construir sobre RAD [N. del T. Desarrollo Rápido de Aplicaciones] y desarrollo iterativo. Habiendo empezado con 17 fundadores ahora tiene más de mil miembros y ha crecido fuera de sus raíces británicas. Siendo desarrollado por un consorcio, tiene un sabor diferente a muchos de los otros métodos ágiles. Tiene una organización de tiempo completo que lo apoya con manuales, cursos de entrenamiento, programas de certificación y demás. También lleva una etiqueta de precio, lo que ha limitado mi investigación sobre su metodología. Sin embargo Jennifer Stapleton ha escrito un libro que da una apreciación global de la metodología.

- **Adaptive Software Development (ASD):** Promovido por Jim Highsmith. Sus principales características son:

- Iterativo.
- Orientado a los componentes software más que a las tareas y tolerante a los cambios.
- El ciclo de vida que propone tiene tres fases esenciales:
  - ✓ Especulación.
  - ✓ Colaboración y aprendizaje.

En la primera de ellas se inicia el proyecto y se planifican las características del software, en la segunda desarrollan las características y finalmente en la tercera se revisa su calidad, y se entrega al cliente. La revisión de los componentes sirve para aprender de los errores y volver a iniciar el ciclo de desarrollo.

- **Feature -Driven Development (FDD):** Define un proceso iterativo que consta de 5 pasos. Las iteraciones son cortas (hasta 2 semanas). Se centra en las fases de diseño e implementación del sistema partiendo de una lista de características que debe reunir el software. Sus impulsores son Jeff De Luca y Peter Coad.

- **Lean Development (LD):** Definida por Bob Charretes en los años 80 y utilizada en numerosos proyectos de telecomunicaciones en Europa. Los cambios se consideran riesgos, pero si se manejan adecuadamente se pueden convertir en

oportunidades que mejoren la productividad del cliente. Su principal característica es introducir un mecanismo para implementar dichos cambios.

- **SCRUM:** Esta metodología se basa en una filosofía del desarrollo ágil, creado por Hirotaka Takeuchi e Ikujiro Nonaka ahí por el año 1986, para desarrollo de software. SCRUM aunque puede ser usado para otro tipo de proyectos y tiene demostrada efectividad en otras áreas, aunque generalmente es funcional solo para desarrollos de software porque para eso fue diseñado. La idea es desarrollar aplicaciones mucho más rápido y eficazmente.

- **XP:** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

### **¿Por qué aplicar Scrum y XP?**

Antes de plantearse un cambio tan importante como el que puede suponer usar una metodología ágil es necesario mirar a tu alrededor y tener claro donde estas y hacia donde quieres ir. Si estas pensando en meterte en el largo (y posiblemente tortuoso) camino de cambiar tu metodología de trabajo lo primero es identificar que problemas tienes y evaluar si una metodología ágil te va a ayudar a resolver estos problemas, si esta metodología encaja en tu organización y es posible aplicarla, si el equipo de desarrollo que las va a aplicar cuenta con los conocimientos y los medios necesarios, vamos que lanzarse a lo loco a usar esto o aquello simplemente porque está de moda es como poco “arriesgado” y podemos terminar consiguiendo que el remedio sea peor que la enfermedad.

Para el desarrollo de este trabajo se propone el uso de dos de las muchas metodologías ágiles enunciadas anteriormente, Scrum para la parte de planificación, y XP para la parte de desarrollo.

### **1.4.1 Scrum**

Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos diez años.

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de treinta días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de quince minutos del equipo de desarrollo para coordinación e integración.

#### **¿Qué significa SCRUM?**

Pues la palabra se usa en rugby y prácticamente significa melé. Si ustedes han visto un juego de rugby cuando va a iniciar una jugada se aglomeran los equipos empujándose con tal de hacerse de la pelota para recuperarla y pasársela a otro que espera atrás. Pues eso es SCRUM no son siglas ni nada por el estilo.

#### **Características de Scrum**

Scrum es un modelo de referencia que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Los roles principales en Scrum son el ScrumMaster, que mantiene los procesos y trabaja de forma similar al director de proyecto, el ProductOwner, que representa a los stakeholders (clientes externos o internos), y el Team que incluye a los desarrolladores.

El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados.

Entonces, se determina la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos están congelados durante el sprint.

Un principio clave de Scrum es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan (a menudo llamado requirements churn), y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, Scrum adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

#### **1.4.2 XP (Extreme Programming)**

XP es una de las metodologías ágiles de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y corto equipo. Centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software. XP se basa en realimentación continua entre el cliente y el desarrollador, comunicación fluida, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas [3].

#### **1.4.3 Valores de la metodología XP**

##### **Simplicidad**

XP propone el principio de hacer la cosa más simple que pueda funcionar, en relación al proceso y la codificación. Es mejor hacer hoy algo simple, que hacerlo complicado y probablemente nunca usarlo mañana.

El programador tiene que estar en constante comunicación con el cliente para satisfacer sus requisitos y responder rápidamente a los cambios de los mismos. Muchos problemas que surgen en los proyectos se deben a que después de concretar los requisitos que debe cumplir el programa no hay una revisión de los mismos, pudiendo dejar olvidados puntos importantes.

### **Comunicación**

El Extreme Programming se nutre del ancho de banda más grande que se puede obtener cuando existe algún tipo de comunicación: la comunicación directa entre personas. Es muy importante entender cuáles son las ventajas de este medio. Cuando dos (o más) personas se comunican directamente pueden no solo consumir las palabras formuladas por la otra persona, sino que también aprecian los gestos, miradas, etc. que hace su compañero. Sin embargo, en una conversación mediante el correo electrónico, hay muchos factores que hacen de esta una comunicación, por así decirlo, mucho menos efectiva. Muchos problemas que surgen en los proyectos se deben a que después de concretar los requisitos que debe cumplir el programa no hay una revisión de los mismos, pudiendo dejar olvidados puntos importantes.

### **Retroalimentación (Feedback)**

Retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales da una mayor oportunidad de dirigir el esfuerzo eficientemente. Mediante la retroalimentación se ofrece al cliente la posibilidad de conseguir un sistema adecuado a sus necesidades. Se le va mostrando el proyecto a tiempo para sugerir cambios y poder retroceder a una fase anterior para rediseñarlo a su gusto.

### **Coraje**

El coraje es un valor muy importante dentro de la programación extrema. Un miembro de un equipo de desarrollo extremo debe de tener el coraje de exponer sus dudas, miedos, experiencias sin “embellecer” éstas de ninguna de las maneras. Esto es muy importante ya que un equipo de desarrollo extremo se basa en la confianza para con sus miembros. Faltar a esta confianza es una falta más que grave. El coraje existe en el contexto de los otros tres valores.

Cada uno de ellos se apoya en los demás:

- Se requiere valor para comunicarse con los demás cuando eso podría exponer la propia ignorancia.
- Se requiere valor para mantener el sistema simple, dejando para mañana las decisiones de mañana.
- Se requiere coraje para confiar en que la retroalimentación durante el camino es mejor que tratar de adivinar todo con anticipación.
- Y, sin un sistema simple, comunicación constante y retroalimentación, es difícil mantenerse valiente.

#### **1.4.4 Prácticas de la metodología XP**

##### **Planificación incremental**

La Programación Extrema asume que la planificación nunca será perfecta, y que variará en función de cómo varíen las necesidades del negocio. Por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de feedback que permitan conocer con precisión dónde estamos. Como es lógico, la planificación es iterativa, un representante del negocio decide al comienzo de cada iteración qué características concretas se van a implementar. El objetivo de la XP es generar versiones de la aplicación tan pequeñas como sea posible, pero que proporcionen un valor adicional claro, desde el punto de vista del negocio. A estas versiones se las denomina releases. Una release cuenta con un cierto número de historias. La historia es la unidad de funcionalidad en un proyecto XP, y corresponde a la mínima funcionalidad posible que tiene valor desde el punto de vista del negocio.

Durante cada iteración se cierran varias historias, lo que hace que toda iteración añada un valor tangible para el cliente. Es fundamental en toda esta planificación la presencia de un representante del cliente, que forma parte del equipo y que decide cuáles son las historias más valiosas. Estas historias son las que se desarrollarán en la iteración actual.

Como se puede ver, y como siempre ocurre con la Programación Extrema, el enfoque utilizado para llevar a cabo la planificación es eminentemente pragmático. Gran parte de la eficacia de este modelo de planificación deriva de una división clara de responsabilidades, que tiene en cuenta las necesidades del negocio en todo momento. Dentro de esta división, el representante del cliente tiene las siguientes responsabilidades:

- Decidir qué se implementa en cada release o iteración.
- Fijar las fechas de fin de la release, recortando unas características o añadiendo otras.
- Priorizar el orden de implementación, en función del valor de negocio.

## **Testing**

La ejecución automatizada de tests es un elemento clave de la XP. Existen tanto tests internos (o tests de unidad), para garantizar que el mismo es correcto, como tests de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir los tests de aceptación, no necesariamente de implementarlos. Él es la persona mejor cualificada para decidir cuál es la funcionalidad más valiosa. El hecho de que los tests sean automatizados es el único modo de garantizar que todo funciona desde el punto de vista de la XP, si no hay tests, las cosas sólo funcionan en apariencia. Aún más, si un test no está automatizado, no se le puede considerar como tal. El objetivo de los tests no es corregir errores, sino prevenirlos. Por ejemplo, los tests siempre se escriben antes que el código a testear, no después: esto aporta un gran valor adicional, pues fuerza a los desarrolladores a pensar cómo se va a usar el código que escriben, poniéndolos en la posición de consumidores del software. Elaborar los tests exige pensar por

adelantado cuáles son los problemas más graves que se pueden presentar, y cuáles son los puntos dudosos. Esto evita muchos problemas y dudas, en lugar de dejar que aparezcan “sobre la marcha”. Un efecto lateral importante de los tests es que dan una gran seguridad a los desarrolladores es posible llegar a hacer cambios más o menos importantes sin miedo a problemas inesperados, dado que proporcionan una red de seguridad. La existencia de tests hace el código muy maleable.

### **Refactorización continúa**

Es el proceso de modificar el código de un sistema de software de modo que no se altere su comportamiento externo pero se mejore su estructura interna. Se refactoriza el sistema eliminando duplicación, mejorando la comunicación y agregando flexibilidad sin cambiar la funcionalidad. El proceso consiste en una serie de pequeñas transformaciones que modifican la estructura interna preservando su conducta aparente. La práctica también se conoce como Mejora Continua de Código o Refactorización implacable. Se lo ha parafraseado diciendo: “Si funciona bien, arréglole de todos modos”.

### **Diseño simple**

Otra práctica fundamental de la Programación Extrema es utilizar diseños tan simples como sea posible. El principio es “utilizar el diseño más sencillo que consiga que todo funcione”. Se evita diseñar características extra porque a la hora de la verdad la experiencia indica que raramente se puede anticipar qué necesidades se convertirán en reales y cuáles no.

La XP nos pide que no vivamos bajo la ilusión de que un diseño puede resolver todas o gran parte de las situaciones futuras lo que parece necesario cambia con frecuencia, es difícil acertar a priori. Es obvio que, si no vamos a anticipar futuras necesidades, debemos poder modificar el diseño si alguna de estas se materializa. La XP soporta estas modificaciones gracias a los tests automatizados. Estos permiten hacer cambios importantes gracias a la red de protección que

proporcionan. La refactorización, que hace que el código existente sea claro y sencillo, también ayuda a hacer factibles las modificaciones.

La XP define un “diseño tan simple como sea posible” como aquél que:

- Pasa todos los tests.
- No contiene código duplicado.
- Deja clara la intención de los programadores (enfatisa el qué, no el cómo) en cada línea de código.
- Contiene el menor número posible de clases y métodos.

### **Propiedad colectiva del código**

La XP aboga por la propiedad colectiva del código. En otras palabras, todo el mundo tiene autoridad para hacer cambios a cualquier código, y es responsable de ellos. Esto permite no tener que estar esperando a otros cuando todo lo que hace falta es algún pequeño cambio. Por supuesto, cada cuál es responsable de las modificaciones que haga. El principio básico es “tú lo rompes, tú lo arreglas, no importa si está en el código propio o en el de otros”. Por último, vale la pena tener en cuenta que la existencia de tests automatizados impide que se produzca un desarrollo anárquico, al ser cada persona responsable de que todos los tests se ejecuten con éxito al incorporar los cambios que ha introducido al programa.

### **Integración continúa**

En muchos casos la integración de código produce efectos laterales imprevistos, y en ocasiones la integración puede llegar a ser realmente traumática, cuando dejan de funcionar cosas por motivos desconocidos. La Programación Extrema hace que la integración sea permanente, con lo que todos los problemas se manifiestan de forma inmediata, en lugar de durante una fase de integración más o menos remota. La existencia de una fase de integración separada tiene dos efectos laterales indeseables se empieza a hacer codificación “yo-yo”, en la que todo el mundo modifica código “sólo para que funcione, ya lo ajustaremos”, y hace que se acumulen

defectos. Evitar que se acumulen defectos es muy importante para la XP, como lo es el conseguir que los defectos que cada programador inyecta los elimine él mismo.

### **Entregas pequeñas**

Siguiendo la política de la XP de dar el máximo valor posible en cada momento, se intenta liberar nuevas versiones de las aplicaciones con frecuencia. Éstas deben ser tan pequeñas como sea posible, aunque deben añadir suficiente valor como para que resulten valiosas para el cliente. Una entrega no debería tardar más tres meses.

### **Estándares de programación**

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible, facilitando los cambios.

## **1.4.5 Fases de la metodología XP**

Hay diversas prácticas inherentes a la Programación Extrema, en cada uno de los ciclos de desarrollo del proyecto.

### **Planificación**

Historias de Usuario: El primer paso de cualquier proyecto que siga la metodología XP es definir las Historias de Usuario con el cliente. Las Historias de Usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias, constan de tres ó cuatro líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base datos adecuados.

Release planning: Tras definir las Historias de Usuario es necesario crear un plan de publicaciones, donde se indiquen las Historias de Usuario que se implementarán para cada versión de la aplicación y las fechas en las que se publicarán dichas versiones.

Iteraciones. Todo proyecto que siga la metodología XP se ha de dividir en iteraciones de aproximadamente 3 semanas de duración. Al comienzo de cada iteración los clientes deben seleccionar las Historias de Usuario definidas en el “Release planning” que serán implementadas.

### **Diseño**

- Diseños simples: La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácil de entender e implementar que a la larga costará menos tiempo y esfuerzo desarrollar.
- Glosarios de términos: Usar una correcta especificación de los nombres de clases, métodos y propiedades ayudará a comprender el diseño y facilitará futuras ampliaciones y la reutilización del código.
- Tarjetas CRC: El uso de las tarjetas CRC (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica.

### **Codificación**

El cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las Historias de Usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los tests que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse ateniendo a estándares y patrones de codificación ya

creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y la escalabilidad. Crear test que prueben el funcionamiento de los distintos códigos implementados nos ayudará a desarrollar dicho código. Crear estos test antes nos ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y sabremos que una vez implementado pasará dichos test sin problemas ya que dicho código ha sido diseñado para ese fin.

### **Pruebas**

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará. Hay que crear los tests abstrayéndose del futuro código, de esta forma aseguraremos la independencia del test respecto al código que evalúa. El uso de los test es adecuado para observar la refactorización. Los test permiten verificar que un cambio en la estructura de un código no tiene por qué cambiar su funcionamiento. Para asegurar el funcionamiento final de una determinada historia de usuario se deben crear “Test de aceptación”; estos tests son creados y usados por los clientes para comprobar que las distintas Historias de Usuario cumplen su cometido.

### **1.5 Conclusiones**

En este capítulo se hace un estudio sobre los procesos penales relacionados a la atención y protección del menor víctima en la Fiscalía Provincial Holguín y se exponen los conceptos necesarios para el entendimiento del software, se toman decisiones importantes, luego de un estudio de las distintas herramientas para el desarrollo de software tales como la elección de los lenguajes de programación y metodologías utilizadas.

## **CAPÍTULO 2: ANÁLISIS Y DISEÑO**

### **2.1 Introducción**

En este capítulo se abordará la fase de planeación y diseño, la cual es muy importante ya que se generan las principales planillas que se encuentran relacionadas con la concepción inicial del sistema. En esta se especificarán los requisitos funcionales y no funcionales del sistema y quedarán plasmadas las historias de usuarios [5].

### **2.2 Planificación – Definición**

La fase de Planificación – Definición es la primera que define la metodología SXP. En esta fase se generan todos los documentos que se encuentran relacionados con la concepción inicial del sistema, así como la definición del mismo. También se incluyen algunos que están vinculados a la primera parte de los procesos de Ingeniería de Software, tales como los relacionados con el negocio, los requisitos y el diseño. Y como también hay una parte de planificación en esta fase, se incluyen aquellos documentos que están relacionados con la estimación inicial de esfuerzos, y la valoración de los riesgos [5].

#### **2.2.1 Concepción del sistema**

La planilla de Concepción del Sistema es el primer documento que se genera en la fase de Planificación-Definición. En este además de reflejarse la visión general del producto a implementar se da a conocer los diferentes roles que intervendrán en el desarrollo del software así como las responsabilidades a asumir en dicho proceso. Se recoge también cuales serán las herramientas utilizadas en la implementación de la aplicación y los motivos de la necesidad del desarrollo del software.

Para comprender cuales son los datos que se recogerán en esta planilla, se muestra la misma a continuación.

### **Clasificación del proyecto**

Aplicación Web.

### **Tipo de proyecto**

Nacional.

### **Resumen**

El presente documento tratará temas relacionados con la atención, protección y control de los menores víctimas en la Fiscalía Provincial Holguín, las razones por las cuales surge el proyecto y qué se pretende solucionar con el mismo, así como los roles que intervendrán en el desarrollo del trabajo por parte de los clientes y desarrolladores.

**Palabras claves:** Trabajo de Diploma.

### **Surgimiento.**

La aplicación acelerada en las últimas décadas de la informática ha permitido en las distintas entidades alcanzar nuevos niveles de desarrollo. La Fiscalía Provincial de Holguín trabaja por incorporar las facilidades y ventajas del uso de las nuevas tecnologías en la vida cotidiana, es por ello que se ha estudiado el proceso de Control y Protección al Menor Víctima, dicho proceso se realiza de forma manual ocasionando que este trabajo sea engorroso para los involucrados en esta tarea por lo que surgió la idea de que era necesario desarrollar una herramienta informática que ayudara en la gestión de estos procesos.

### **¿Qué es?**

La Automatización de Procesos Penales relacionados con la Protección y Control al Menor Víctima no es más que una vía de solución para gestionar y controlar de forma segura y ágil toda la información referente a estos menores que han sido víctimas de algún hecho y que están en proceso de seguimiento por la Fiscalía Provincial Holguín.

### **Metodología a utilizar.**

SCRUM-XP (SXP).

### **Roles.**

**Tabla 2.1 Roles que intervienen en el desarrollo del sistema**

<b>Rol</b>	<b>Nombre</b>
Gerente	Ing. Junier García Nicot
Analista	Javier Calzadilla López
Diseñador	Javier Calzadilla López
Desarrollador	Javier Calzadilla López
Cliente	Fiscalía Provincial Holguín

### **Misión**

La aplicación web que se desea desarrollar tiene como misión constituir una herramienta básica y fundamental que permita a los fiscales gestionar y hacer un seguimiento de forma ágil de la información concerniente con los Menores Víctimas.

### **Visión.**

Se espera contar con una aplicación web robusta e interactiva, donde el usuario pueda nutrirse e interactuar con la información y los servicios con los que desea contar, que permita un rápido y seguro acceso a estos cumpliendo con las expectativas.

### **Tecnologías y herramientas utilizadas**

- Metodología SXP.
- PHP.
- MYSQL.
- Dreamweaver.

### **2.2.2 Modelo de Historias de usuario del negocio**

Mediante el Modelo de Historias de usuario del negocio se concretan las características específicas del negocio, así como la forma en que interactúa el sistema con los clientes y viceversa. Como se está trabajando con metodologías ágiles, el modelo de negocio es diferente al ya conocido en el proceso unificado, ya que en este caso se trabaja con historias de usuarios, en vez de con casos de uso.

Pero independientemente de los cambios técnicos que puedan existir, el negocio se modela igual en cualquier metodología.

A continuación se muestran cuales son los actores que intervienen en el negocio.

**Tabla 2.2 Actores del negocio**

<b>Actor</b>	<b>Descripción</b>
Supervisor Provincial	Tendrá el derecho de crear los distintos usuarios del sistema y visualizar información necesaria.
Fiscales	Son los encargados de manejar el flujo de información requerida en el sistema.

### 2.2.3 Lista de reserva del producto (LRP)

La planilla de Lista de reserva del producto es el primer artefacto generado en la etapa de captura de requisitos y está conformada por una lista priorizada que define el trabajo que se va a realizar en el proyecto. Cuando un proyecto comienza es difícil tener bien claros todos los requerimientos del producto. Se crea una lista de requerimientos la cual puede crecer y modificarse a medida que se obtienen más conocimientos acerca del producto y del cliente, con la restricción de que sólo puede cambiarse entre iteraciones. El objetivo es asegurar que el producto definido al terminar la lista es el más correcto, útil y competitivo posible y para esto la lista debe acompañar los cambios en el entorno y el producto.

Para comprender cuales son los datos que se recogerán en esta planilla, se muestra la misma gráficamente.

**Tabla 2.3 Lista de reserva del producto**

<b>Asignado a</b>	<b>Ítem *</b>	<b>Descripción</b>	<b>Estimación</b>	<b>Estimado por</b>
	<b>Prioridad</b>		<b>Muy Alta</b>	
Javier Calzadilla	1	El supervisor crea los usuarios	1	Analista

*Automatización de Procesos Penales.  
Javier Calzadilla López.*

López		del sistema		
Javier Calzadilla López	2	Los fiscales crean los expedientes	1	Analista
Javier Calzadilla López	3	El supervisor revisa y controla los expedientes creados	1	Analista
<b>Prioridad</b>			<b>Muy Alta</b>	
Javier Calzadilla López	4	El sistema emite una advertencia de los expedientes a ser revisados al cumplimiento de su término	3	Analista
Javier Calzadilla López	5	El supervisor determina el curso del expediente	3	Analista
<b>Prioridad</b>			<b>Baja</b>	
<b>Requisitos no Funcionales</b>				
Javier Calzadilla López	6	<b>Apariencia o Interfaz Externa</b> Diseño sencillo permitiendo que se pueda interactuar de manera fácil con el sistema. Su interfaz agradable logra captar la confianza de los usuarios así como su identificación con los colores y formatos en que realizan sus acciones	14 semanas	
Javier Calzadilla López	7	<b>Rendimiento</b> Como la aplicación está concebida para un ambiente cliente/servidor los tiempos de respuestas deben ser generalmente rápidos al igual que la velocidad de procesamiento de la información. Estará implementado sobre una tecnología web, facilitando su uso a través de la red	14 semanas	
Javier Calzadilla López	8	<b>Soporte</b> La base de datos que utiliza el sistema como medio de almacenamiento de la información está soportada sobre un gestor de bases de datos MYSQL	14 semanas	
Javier Calzadilla López	9	<b>Usabilidad</b> El sistema en general se desarrolla con el objetivo de	14 semanas	

		facilitar el trabajo manual realizado hasta el momento por lo que se espera cuente con un alto nivel de usabilidad por los usuarios debido a que constituye una forma más fácil de realizar el trabajo. El sistema garantizará un acceso fácil y rápido por los usuarios		
Javier Calzadilla López	10	<b>Portabilidad</b> El sistema trabaja sobre Windows y Linux	14 semanas	

#### 2.2.4 Historias de usuario

Las historias de usuario son la técnica utilizada en XP para especificar los requisitos del software, lo que equivaldría a los casos de uso en el proceso unificado. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción depende principalmente de la habilidad que tenga el cliente para definir las. Las historias de usuario guían la construcción de las pruebas de aceptación, elemento clave en XP y son utilizadas para estimar tiempos de desarrollo. En este sentido, sólo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas. En el momento de implementar una historia de usuario, se debe detallar a través de la comunicación con el cliente. Estas son la base para las pruebas funcionales.

Para comprender cuales son los datos que se recogerán en esta planilla, se muestra la misma gráficamente.

**Tabla 2.4 Descripción de la Historia de Usuario**

Historia de Usuario	
<b>Número: 1</b>	<b>Nombre Historia de Usuario:</b> Gestionar Usuario.
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Javier Calzadilla López	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 3

<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2.5
<b>Descripción:</b> Esta Historia de Usuario es la que permite el tratamiento a los usuarios del sistema.	
<b>Observaciones:</b> Se hace referencia a la funcionalidad 1 enumerada en la Lista de reserva del producto.	

Para ver las demás planillas de Historias de Usuario, ir al Expediente de Proyecto 04-Planilla Historias de Usuarios.

### 2.2.5 Valoración de riesgos

La planilla de lista de riesgos, es el documento que se genera de la actividad de valoración de riesgos. En ella se definen los posibles riesgos que actuarán sobre el proceso de desarrollo de software, así como la estrategia trazada para mitigarlos, además de un plan de contingencia que describe que curso seguirán las acciones si el riesgo se materializa.

Para comprender cuales son los datos que se recogerán en esta planilla, se muestra la misma gráficamente.

**Tabla 2.5 Lista de riesgo**

<b>Riesgos</b>	<b>Tipos de riesgos</b>	<b>Impacto</b>	<b>Descripción</b>	<b>Probabilidad</b>	<b>Efectos</b>	<b>Mitigación de riesgos</b>
1. Escaso dominio de las herramientas a utilizar.	Tecnológico	Puede atrasar el plazo de entrega del software.	Estudiando las herramientas y tecnologías a utilizar en el desarrollo del software se puede llegar a la conclusión que estas son las	Medias.	Serias.	1. Intenso estudio de las bibliografías que tratan estas tecnologías. 2. Entrevistas con personas capacitadas en

*Automatización de Procesos Penales.  
Javier Calzadilla López.*

			más factibles a emplear por sus características, muchas de ellas son nuevas, es decir, que se tiene poco dominio de su uso por lo que se considera un riesgo para la investigación.			el tema.
2. Falta de comunicación con los clientes.	Organización.	Pobre captura de requisito.	Los clientes tienen mucha responsabilidad por lo que no disponen de mucho tiempo para realizar reuniones y así determinar los requisitos del sistema.	Media.	Serias.	1. Planificación de reuniones previstas con los clientes.

**Plan de Contingencia.**

**• Riesgo # 1**

En caso de que no se logre mitigar este riesgo se debe tomar como una solución alternativa utilizar herramientas y tecnologías que se conozcan.

**• Riesgo # 2**

Si no se logra el contacto con los clientes se deberá considerar estudiar el negocio por otras vías.

### **2.2.6 Modelo de diseño**

La planilla del Modelo de diseño, es el documento que se genera del Diseño con las metáforas, donde se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume de forma evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. Teniendo en cuenta las características anteriores, se define un esbozo inicial del diseño del sistema, sin entrar en especificaciones, ni detalles, sólo lo que se necesita para hacer un primer entregable del sistema.

Rol: Diseñador.

### **2.3 Conclusiones del capítulo**

Con el desarrollo de este capítulo se pudo comprender mejor la interacción del sistema con los usuarios. Se lograron definir de forma organizada todos los requisitos, tanto funcionales como no funcionales, los cuales sirvieron de guía en el proceso de implementación. También se logró precisar los posibles riesgos para el proyecto, así como la forma de mitigarlos, lo cual disminuye el efecto de los mismos, si ocurrieran.

## **CAPÍTULO 3: DESARROLLO DEL SISTEMA**

### **3.1 Introducción**

En este capítulo se presenta la fase de desarrollo de la metodología utilizada. En esta se describen cada unas de las tareas asociadas a cada historia de usuario y además se realiza el cronograma de producción en el cual se refleja cada una de las actividades a desarrollar por el equipo de trabajo.

### **3.2 Desarrollo**

La fase de Desarrollo, es la segunda que define la metodología XP - Scrum. En la primera parte de esta fase se generan todos los documentos relacionados con la planificación de las iteraciones así como de las tareas a realizar durante la implementación [5].

#### **3.2.1 Plan de releases**

En este paso se define el plan de releases e iteraciones para realizar las entregas intermedias y la entrega final. Tiene como entrada la relación de historias de usuario definidas previamente. Para colocar una historia en cada iteración se tuvo en cuenta la prioridad de dicha historia. Como resultado de la priorización de historias se llegó a la siguiente planificación:

**Tabla 3.1 Plan de realeses**

<b>Release</b>	<b>Descripción de la Iteración</b>	<b>Orden de la HU a implementar</b>	<b>Duración total</b>
1	En esta iteración se implementará una de las historias de usuarios más	1	3

	importante del sistema ya que tiene que ver con la gestión de usuario.		
2	En esta iteración se implementará la historia de usuario que tiene que ver con la gestión de registros para conformar los expedientes.	2	3
3	En esta iteración se implementará la historia de usuario que tiene que ver con la alarma ya que esta es muy importante dentro del sistema.	3.5	3
4	En esta iteración se implementará la historia de usuario listar información, HU muy importante para el cliente dentro del sistema.	4	3

### 3.2.2 Cronograma de Producción

La planilla de Cronograma de Producción, es un documento de suma importancia, pues fue el que guió y controló las actividades que se realizaron en cada iteración. En éste, no sólo se recogen las actividades planificadas, sino el tiempo que durarán y el rol responsable de desarrollarlas.

Para comprender cuales son los datos que se recogerán en esta planilla, se muestra la misma gráficamente.

**Tabla 3.2 Cronograma de Producción**

No	Hito	Descripción	Inicio	Fin	% Ejec	Ejecutor
----	------	-------------	--------	-----	--------	----------

*Automatización de Procesos Penales.  
Javier Calzadilla López.*

1	Entrevista fiscales de la Fiscalía Provincial Holguín.	Se desarrollaron entrevistas con el objetivo de conocer como se lleva a cabo el proceso control, atención y protección al menor víctima.	20/12/2010	27/12/2010	100	Analista
2	Estudio de herramientas utilizar en desarrollo de aplicación.	Esta actividad se realizó para poder conocer con mayor profundidad las herramientas a utilizar en la implementación del software.	10/01/2011	15/02/2011	95	Desarrollador
3	Diseño de la Base Datos.	Se diseñaron las tablas que se necesitarían para el manejo de la información.	15/02/2011	16/02/2011	100	Diseñador
4	Diseño.	Se diseñó la interfaz de usuario.	17/02/2011	25/02/2011	95	Diseñador
5	Implementación.	Se implementó	08/02/2011	28/02/2011	100	Desarrollador

		la lógica del negocio.				
6	Pruebas.	Se realizaron pruebas a las diferentes historias de usuario.	02/03/2011	23/03/2011	98	Desarrollador
7	Factibilidad.	Se realizó un estudio de factibilidad.	20/05/2011	22/05/2011	100	Analista

### 3.2.3 Tarjetas CRC

Las tarjetas CRC forman parte de las técnicas propuestas por algunos de los creadores de la metodología ágil XP (Ward Cunningham y Kent Beck), con el objetivo de obtener un diseño simple y que no incurra en la implementación de funcionalidades que no son necesarias. Esta técnica de modelado permite entender las características del sistema pensando en términos de objetos y clases [5].

Las tarjetas CRC permiten que el equipo completo contribuya en la tarea del diseño. Una tarjeta CRC representa un objeto. El nombre de la clase se coloca a modo de título en la tarjeta, los atributos y las responsabilidades más significativas se colocan a la izquierda, y las clases que están implicadas en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

Debido a la gran facilidad de uso y entendimiento, que propician dichas tarjetas, el equipo de desarrollo del presente trabajo, decidió utilizarlas para diseñar el sistema que se desea desarrollar.

### 3.2.4 Tareas por Historias de Usuario

La planilla de Tarea de ingeniería, posee una gran importancia, pues permite definir cada una de las actividades que estarán asociadas a las historias de usuario y que permitirán su implementación. También posibilita conocer que programador está asignado a cada tarea, así como el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo que se llevará cada historia de usuario en implementarse, de acuerdo a su complejidad.

Para comprender cuales son los datos que se recogerán en esta planilla, se muestra la misma gráficamente.

**Tabla 3.3 Descripción de la tarea de ingeniería**

<b>Tarea de Ingeniería</b>	
<b>Número Tarea: 1</b>	<b>Número Historia de Usuario: 1</b>
<b>Nombre Tarea:</b> Crear Usuario	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 30/01/2011	<b>Fecha Fin:</b> 31/01/2011
<b>Programador Responsable:</b> Javier Calzadilla López	
<b>Descripción:</b> Un usuario es creado por el administrador del sistema el cual le da un nivel de acceso.	

Para ver las demás Planillas de Tareas de Ingeniería, ir al Expediente de Proyecto: 07- Tarea de Ingeniería.

### **3.3 Conclusiones del capítulo**

En este capítulo se confeccionaron las tarjetas CRC. Además se dividieron en tareas de ingeniería cada una de las historias de usuario para poder lograr la completa implementación de cada una en la fecha acordada con el cliente, lo cual posibilitó conocer el grado de complejidad de estas teniendo en cuenta la cantidad de tareas asociadas. A cada tarea de ingeniería se le asignó un tiempo de desarrollo que se cumplió de manera eficiente garantizando así el objetivo principal de su confección.

## **CAPÍTULO 4: PRUEBAS**

### **4.1 Introducción**

En el presente capítulo se muestran las pruebas de aceptación elaboradas por el cliente para demostrar que la aplicación funcione de forma correcta. Estas pruebas fueron llevadas a cabo antes de cada entrega que se realizó durante todo el desarrollo del proyecto.

### **4.2 Pruebas**

Una de las mejores características de la metodología XP es el proceso de pruebas. Esta metodología propone probar tanto como sea posible. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos no deseados a la hora de realizar modificaciones y refactorizaciones. XP propone la realización de pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente [5].

#### **4.2.1 Pruebas de aceptación**

En esta planilla se escriben las pruebas ejecutadas según la historia de usuario seleccionada para realizar la comprobación y validar las funcionalidades del sistema, y de esta forma saber si está apto para ser liberado. A continuación se muestran un Caso de Prueba de Aceptación, el cual define en cada uno su evaluación final.

**Tabla 4.1 Caso de prueba de aceptación**

<b>Caso de Prueba de Aceptación</b>	
<b>Código de Caso de Prueba:</b> APP-1-1	<b>Nombre de Historia de usuario:</b> Gestionar Usuario
<b>Nombre de la persona que realiza la prueba:</b> Javier Calzadilla López	
<b>Descripción de la prueba:</b> El supervisor gestiona los usuarios. (Registrar Usuario, Cambiar nivel de acceso, Cambiar Clave, Eliminar Usuario)	
<b>Condiciones de ejecución:</b> Para crear un usuario se deben llenar todos los campos.	
<b>Entrada / Pasos de ejecución:</b> El supervisor inserta todos los datos.	
<b>Resultado esperado:</b> El sistema debe procesar toda la información sin mostrar ningún error.	
<b>Evaluación de la prueba:</b> Satisfactorio.	

Para ver las demás Planillas Caso de prueba de aceptación, ir al Expediente de Proyecto: 09- Planillas Caso de prueba de aceptación.

### **4.3 Conclusiones del capítulo**

Con la realización de las pruebas de aceptación se asegura de que las funciones implementadas cumplan su objetivo satisfactoriamente, probando individualmente cada historia de usuario y asignándole la evaluación correspondiente. Todas las pruebas que se realizaron fueron positivas y se estuvo conforme, cumpliendo entonces la aplicación con la definición inicial de las historias de usuarios.

## **CAPÍTULO 5: ESTUDIO DE FACTIBILIDAD**

### **5.1 Introducción**

Para estudiar la factibilidad de este proyecto se utilizará la Metodología Costo-Efectividad (Beneficio), la cual plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo, que involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware/software y los costos de operación asociados
- La efectividad, que se entiende como la capacidad del proyecto para satisfacer la necesidad, solucionar el problema o lograr el objetivo para el cual se ideó, es decir, un proyecto será más o menos efectivo con relación al mayor o menor cumplimiento que alcance en la finalidad para la cual fue ideado (costo por unidad de cumplimiento del objetivo).

Esta tiene como objetivo fundamental proporcionar una medida de los costos que incurren en la realización de un proyecto informático, y a su vez comparar dichos costos previstos con los beneficios esperados de la realización de dicho proyecto [5].

### **5.2 Efectos Económicos**

- Efectos directos.
- Efectos indirectos.
- Efectos externos.
- Intangibles.

#### **5.2.1 Efectos directos**

##### **Efectos positivos**

- Ahorro de tiempo para cumplir con las tareas a desarrollar en el proceso.

- Facilita la información necesaria a los estudiantes y profesores de los distintos departamentos implicados en el proceso.
- Permite a los usuarios realizar los cambios pertinentes en la propuesta inicial de los temas de tesis.

### **Efectos negativos**

- Para usar la aplicación es indispensable el uso de un ordenador, aparejado a los gastos que este trae de consumo de corriente eléctrica y mantenimiento.

### **5.2.2 Efectos indirectos**

Los efectos económicos observados que pudiera repercutir sobre otros mercados no son perceptibles, aunque este proyecto no está construido con la finalidad de comercializarse.

### **5.2.3 Efectos externos**

Se contará con una herramienta adecuada que permitirá a los usuarios gestionar y visualizar la información correspondiente al proceso control, atención y protección al menor víctima.

### **5.2.4 Intangibles**

En la estimación financiera siempre hay elementos visibles por un grupo como perjudiciales o beneficiosos, pero al momento de darle valor en unidades monetarias esto resulta difícil o prácticamente imposible. A fin de medir con precisión los efectos, deberán considerarse dos situaciones:

### **Situación sin Proyecto**

Teniendo en cuenta esta situación el fiscal debe de realizar el trabajo de forma manual lo que retrasa en gran manera el trabajo. Por otro lado a la hora de buscar cierta información solicitada por el supervisor se le hace un tanto engorrosa ya que existe un gran volumen de esta debido a que nada se desecha o se elimina si no que se va archivando.

### **Situación con Proyecto**

Mediante este sistema es posible gestionar y controlar todo el flujo de información concerniente con los menores víctimas, de esta forma se agiliza en gran medida el trabajo de los clientes y queda garantizado la seguridad de los datos accediendo a ellos con facilidad y rapidez.

### **5.3 Beneficios y Costos Intangibles en el Proyecto**

#### **Costos:**

- Resistencia al cambio.

#### **Beneficios:**

- Óptimo aprovechamiento del tiempo a la hora de crear un expediente.
- Óptimo aprovechamiento del tiempo en la búsqueda de información.
- La tecnología utilizada es totalmente libre y gratuita por lo que no se incurre en gastos referentes al pago de licencias.
- Mejora en la calidad y visibilidad de la información.
- Mayor comodidad para todos los involucrados en el proceso.

### **5.3 Ficha de costo**

Para determinar el costo económico del proyecto se utilizará el procedimiento para elaborar una Ficha de Costo de un Producto Informático. Para la elaboración de la ficha se consideran los siguientes elementos de costo, desglosados en moneda libremente convertible y moneda nacional. [2]

**Costos en Moneda Librementemente Convertible:**

• **Costos Directos.**

1. Compra de equipos de cómputo: No procede.
2. Alquiler de equipos de cómputo: No procede.
3. Compra de licencia de Software: No procede.
4. Depreciación de equipos: \$ 25.00
5. Materiales directos: No procede.

**Total: \$ 25.00 CUC**

• **Costos Indirectos.**

1. Formación del personal que labora el proyecto: No procede.
2. Gastos en llamadas telefónicas: No procede.
3. Gastos para el mantenimiento del centro: No procede.
4. Know How: No procede.
5. Gastos en representación: No procede.

**Total: \$0.00.**

• **Gastos de distribución y venta.**

1. Participación en ferias o exposiciones: No procede.
2. Gastos en transportación: No procede.
3. Compra de materiales de propagandas: No procede.

**Total: \$0.00.**

**Costos en Moneda Nacional:**

• **Costos Directos.**

1. Salario del personal que laborará en el proyecto: \$100.00
2. El 5% del total de gastos por salarios se dedica a la seguridad social: No procede.
3. El 0.09% de salario total, por concepto de vacaciones a acumular: No procede

4. Gasto por consumo de energía eléctrica: \$ 340.00
5. Gastos en llamadas telefónicas: No procede.
6. Gastos administrativos: No procede.

• **Costos Indirectos.**

1. Know How: No procede

**Total: \$ 440.00 MN**

Como se hizo referencia anteriormente, la técnica seleccionada para evaluar la factibilidad del proyecto es la Metodología Costo-Efectividad. Dentro de esta metodología la técnica de punto de equilibrio aplicable a proyectos donde los beneficios tangibles no son evidentes, el análisis se basa exclusivamente en los costos. Para esta técnica es imprescindible definir una variable discreta que haga variar los costos. Teniendo en cuenta que el costo para este proyecto es despreciable, se tomará como costo el tiempo empleado por un fiscal para contactar al supervisor (Fiscal Provincial) y entregarle el expediente para que lo revise dentro del término del mismo.

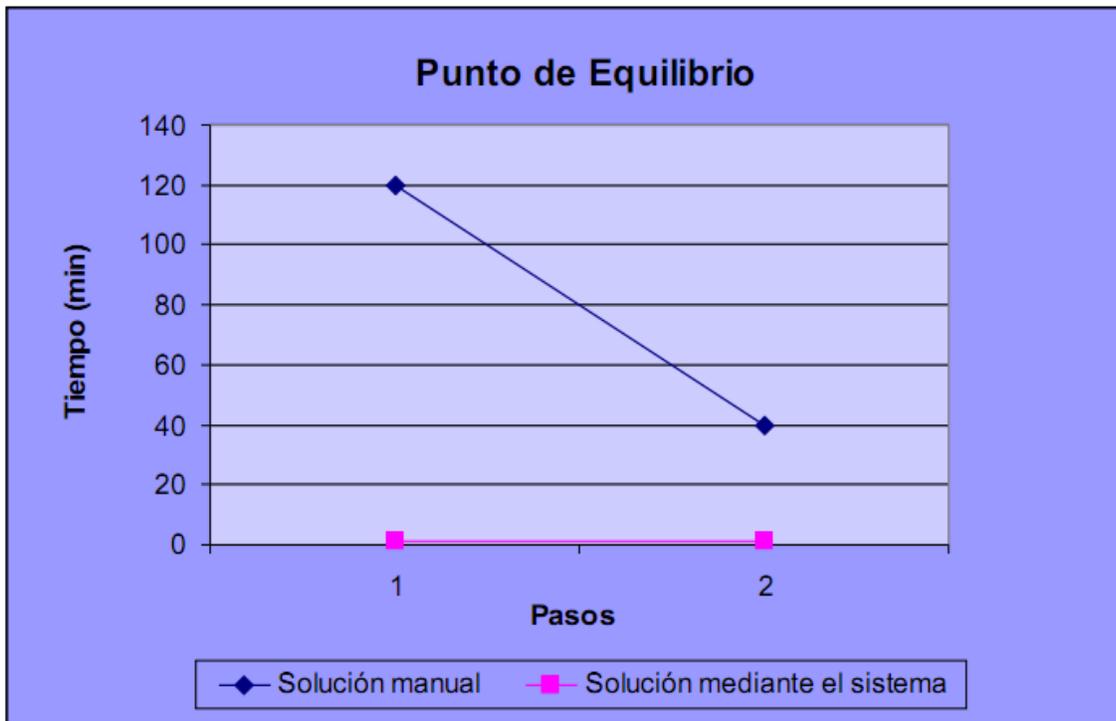
Para ello se ha dividido el proceso de la siguiente manera:

**Valores de la variable (Solución manual)**

1. El fiscal crea un expediente a partir de la denuncia que le llega. (120 min.)
2. El fiscal contacta al supervisor (Fiscal Provincial) y da parte. (30 min.)
3. El fiscal le entrega al supervisor (Fiscal Provincial) el expediente para que lo revise. (10 min.)

**Valor de la variable (Solución mediante el sistema)**

1. El fiscal inserta en el sistema el expediente creado. (1 min.)
2. El supervisor (Fiscal Provincial) revisa en el sistema el expediente creado por el fiscal. (1 min.)



**Figura 5.1 Gráfico de punto de equilibrio de soluciones**

Para lograr una mejor comprensión del gráfico mostrado se debe explicar que algunas de las soluciones fueron concatenadas para lograr una semejanza con la solución mediante el sistema, de forma tal que:

**Tabla 5.1 Explicación del gráfico de punto de equilibrio de soluciones**

	<b>Solución Manual</b>	<b>Solución mediante el sistema</b>
<b>Paso 1</b>	El supervisor solicita información	El fiscal publica en el sistema la información a revisar por el supervisor
<b>Paso 2</b>	El fiscal entrega al supervisor la información	El supervisor visualiza en el sistema la información que necesita

#### **5.4 Conclusiones del capítulo**

En este capítulo se realizó el estudio de factibilidad del proyecto mediante la Metodología Costo-Efectividad (Beneficio). En este se analizaron los efectos económicos, los beneficios y costos intangibles, y además se calculó el costo de ejecución del proyecto mediante la ficha de costo, la cual arrojó como resultado \$ 25.00 CUC y \$ 440.00 MN demostrándose la factibilidad del proyecto.

## **CONCLUSIONES GENERALES**

Con el desarrollo de este proyecto se logró obtener un sistema informático seguro, confiable y de fácil utilización pues trata de no distanciarse mucho de la forma de realización actual de los procesos en cuanto a la estructura y organización del trabajo.

Una vez concluida la investigación e implementación del sistema, se dieron cumplimiento a los objetivos planteados, obteniéndose los siguientes resultados:

- El estudio de cómo se llevan a cabo los procedimientos penales en la esfera de atención y protección al menor víctima permitieron tener una visión más amplia del mismo, resultando más fácil su comprensión para el desarrollo del sistema.
- La realización de un estudio de las herramientas y tecnologías utilizadas permitieron avanzar más rápido en la implementación de la aplicación.
- Se realizó todo el proceso de desarrollo del software siguiendo las fases de la metodología SXP, en las que quedaron plasmadas las tarjetas CRC, se desarrollaron las tareas correspondientes para dar solución a las historias de usuario y se realizaron las pruebas de aceptación las cuales demostraron la conformidad del cliente.
- Se desarrolló una aplicación Web dinámica que permitirá el tratamiento y control de la información en la Fiscalía Provincial Holguín.

## **RECOMENDACIONES**

- Perfeccionar e incorporar nuevas funcionalidades que ayuden a ganar en eficiencia la solución propuesta.
- Incentivar el estudio de nuevas herramientas para proporcionarle mayor funcionalidad a la aplicación web.
- Ampliar la visión de la aplicación.

## **REFERENCIAS BIBLIOGRÁFICAS**

- [1] DÍAZ COBAS, W. *Implementación del módulo Caja del Sistema Integral de Gestión CEDRUX*. Trabajo de Diploma. UCI. 2009.
- [2] GARCÍA PERÉZ, A M. *Procedimiento para la elaboración de la ficha de costo de un producto informático*. Facultad MFC UCLV. Villa Clara
- [3] JEFFRIES, CANOS LETELIER P, FERRER, C. “*Extreme Programming*”, Addison-Wesley. 2001.
- [4] LAMAS LUPERÓN, L M. *Sistema de Control de Estudiantes Becados para la Residencia Estudiantil en el ISMM*. Trabajo de Diploma, ISMM, 2009.
- [5] NICOT HERNÁNDEZ, V. *Sistema de pregrado: Módulo de gestión y seguimiento de los trabajos de diplomas*. ISMM. Moa. 2010.

## **BIBLIOGRAFÍAS**

- BASULTO, J M. *Sistema de Gestión integral de la empresa Empleadora del NÍQUEL "EMPLÉN". MÓDULO DE GESTIÓN DE CONTRATOS DE COMPRAS.* ISMM. Moa. 2009
- Desarrollo ágil de software.* [en línea]. [Consultado 15-03-2011]. Disponible en: [http://es.wikipedia.org/wiki/Desarrollo\\_%C3%A1gil\\_de\\_software](http://es.wikipedia.org/wiki/Desarrollo_%C3%A1gil_de_software)
- DÍAZ COBAS, W. *Implementación del módulo Caja del Sistema Integral de Gestión CEDRUX.* Trabajo de Diploma. UCI. 2009.
- GARCÍA MONTES, M.; GONZÁLEZ RICARDO, O. *Aplicación de las TIC en la Educación Superior.* [en línea]. [Consultado: 2011-04-8]. Disponible en: <http://www.monografías.com/>
- GARCÍA PERÉZ, A M. *Procedimiento para la elaboración de la ficha de costo de un producto informático.* Facultad MFC UCLV. Villa Clara.
- HERNÁN RUIZ, M. *Programación Web Avanzada.* La Habana: Editorial Félix Varela. 2006
- JEFFRIES, CANOS LETELIER P, FERRER, C. *"Extreme Programming"*, Addison-Wesley. 2001.
- LAMAS LUPERÓN, L M. *Sistema de Control de Estudiantes Becados para la Residencia Estudiantil en el ISMM.* Trabajo de Diploma, ISMM, 2009.
- LETELIER, P.; M<sup>a</sup> C. PENADES. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* [en línea]. [Consultado 25-02-2011]. <http://www.willydev.net/descargas/masyxp.pdf>

NICOT HERNÁNDEZ, V. *Sistema de pregrado: Módulo de gestión y seguimiento de los trabajos de diplomas.* ISMM. Moa. 2010.

MATOS GARCÍA, R M. *Sistema de Bases de Datos.* Villa Clara.2006.

OROZCO SILVA, E, GARCÍA DÍAZ, I. *Del dato a la decisión: la gestión de información en un sector específico.* Caso de estudio BIOTEC. Ciencias de la Información 1992.

PONJUÁN DANTE, G. *Gestión de información en las organizaciones: principios, conceptos y aplicaciones.* Santiago de Chile: Cecapi, 1998. pp. XXII.

SCHEIN, E H. *Process consultation.* Cambridge: Addison-Wesley Publishing Company, 1988. pp. 81.

## **GLOSARIO DE TÉRMINOS**

**Apache:** Servidor de páginas Web de código abierto para diferentes plataformas (UNIX, Windows, etc.). **CMS (Content Management System):** Sistema que facilita la gestión de contenidos en todos sus aspectos: creación, mantenimiento, publicación y presentación. También se conoce como Web Content Management (WCM) sistema de gestión de contenido de webs.

**Artefacto:** En tecnología, es un dispositivo concebido y fabricado, sea de modo artesanal o industrial, por una o más personas.

**Calidad:** La palabra calidad tiene múltiples significados. La calidad de un producto o servicio es la percepción que el cliente tiene del mismo. Es una fijación mental del consumidor que asume conformidad con un producto o servicio determinado, que solo permanece hasta el punto de necesitar nuevas especificaciones. La calidad es un conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas.

**Ciclo de vida:** Es un proceso por el cual los analistas de sistemas, los ingenieros de software, los programadores y los usuarios finales elaboran sistemas de información y aplicaciones informáticas.

**Cliente:** Persona, organización o grupo de personas que encargan la construcción de un producto software.

**Desarrollo incremental:** Forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema. Es una combinación del Modelo de Cascada y Modelo Evolutivo.

**Herramientas:** Son los ambientes de apoyo necesario para automatizar las prácticas de Ingeniería de Software.

**Herramienta CASE:** Computer Aided Software Engineering, son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores.

**Imagen:** Grupo de configuraciones específicas que se extraen de una PC para distribuir las en un grupo de PC.

**Iteraciones:** En el contexto de un proyecto se refieren a la técnica de desarrollar y entregar componentes incrementales de funcionalidades de un negocio. Una iteración resulta en uno o más paquetes atómicos y completos del trabajo del proyecto que pueda realizar alguna función tangible del negocio. Múltiples iteraciones contribuyen a crear un producto completamente integrado.

**Libre:** O Software Libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

**Metodología ágil:** Nuevo enfoque metodológico orientado a la gente y los resultados.

**Metodología de desarrollo:** Es una versión amplia y detallada de un ciclo de vida completo de desarrollo de sistemas que incluye: reglas,

procedimientos, métodos, herramientas, funciones individuales y en grupo por cada tarea, productos resultantes, normas de Calidad.

**Metodologías tradicionales:** Metodologías basadas en procesos.

**PHP (Hypertext Pre-processor):** Lenguaje de programación para el desarrollo de webs dinámicas, con sintaxis parecida a la de C. Originalmente se conocía como Personal Home Page tools, herramientas para páginas personales (en Internet).

**Procedimiento:** Son los mecanismos de gestión que soportan a los métodos: El control de los proyectos, el control de la calidad.

**Proceso:** secuencia de actividades que tienen un marcado inicio y fin.

Proyecto de desarrollo: Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

**Protocolo:** Conjunto de normas que rigen un determinado proceso de comunicación.

**Pruebas de aceptación:** son las pruebas realizadas por el cliente para validar el software.

**Requisitos:** Capacidades, condiciones o cualidades que el sistema debe cumplir y tener.

**Servidor:** Computadora central de un sistema de red que provee servicios y recursos (programas, comunicaciones, archivos, etc.) a otras computadoras (clientes) conectadas a ella.

**TCP/IP:** Transfer Control Protocol / Internet Protocol. Son los protocolos que se utilizan en Internet para transmitir datos. El TCP está orientado a la conexión que establece una línea de diálogo entre el emisor y el receptor antes de que se transfieran los datos.

El IP trata cada paquete de forma independiente e incluye en la cabecera información adicional para así controlar la información. Estos protocolos garantizan que la comunicación entre dos aplicaciones es precisa.

**Validación:** no es más que verificar que un producto determinado cumple con los requisitos que fueron pactados con el cliente.