DEDICATORIA

Dedico este proyecto a mis padres y hermana por el apoyo que me han brindado a lo largo de toda mi vida estudiantil ya que con su ayuda y consejos han sabido guiarme para ser una persona de bien tanto en el aspecto humano como en el profesional.

Ellos han sido mi ejemplo a seguir para lograr ser una persona correcta, honesta y con objetivos claros.

Raúl Andrés Landázuri A

Dedico este trabajo principalmente a mis padres los cuales me han ayudado en todas las etapas de mi vida siendo ellos mi motivación para poder superar todos los obstáculos que se me han presentado a lo largo de mi carrera lo cual ha sido muy importante.

Richard Vinicio Siza H

I CUERPO DEL TRABAJO

1. Tema

“Diseño, desarrollo e implementación del sistema de gestión de turnos programados para el Sub-centro de Salud Carapungo**”.**

2. Datos de la organización o institución

Nombre de la Unidad Operativa: Sub-centro de Salud Carapungo

Actividad: Empresa estatal que da atención medica ambulatoria y emergencia.

Ubicación: Provincia de Pichincha Cantón Quito

Características: El sub-centro de salud es una institución que pertenece al ministerio de salud pública del Ecuador que se dedica a brindar atención médica.

Contexto: El centro médico posee las áreas de Medicina Familiar
Medicina General, Ginecología, Obstetricia, Odontopediatría, Odontología, Enfermería, Vacunas, Laboratorio Clínico y Farmacia.

3. Organigrama

En general las empresas de salud tienden a establecer una organización horizontal siguiendo un modelo "arquitectónico". De ésta manera, desde el punto de vista jerárquico y a modo de ejemplo, el organigrama quedaría definido de ésta manera:

Autores: Landázuri Raúl, Siza Richard.
Fecha: 04 – 04 – 2010
Descripción: Organigrama funcional del SCS Carapungo #2.
Fuente: SCS Carapungo #2

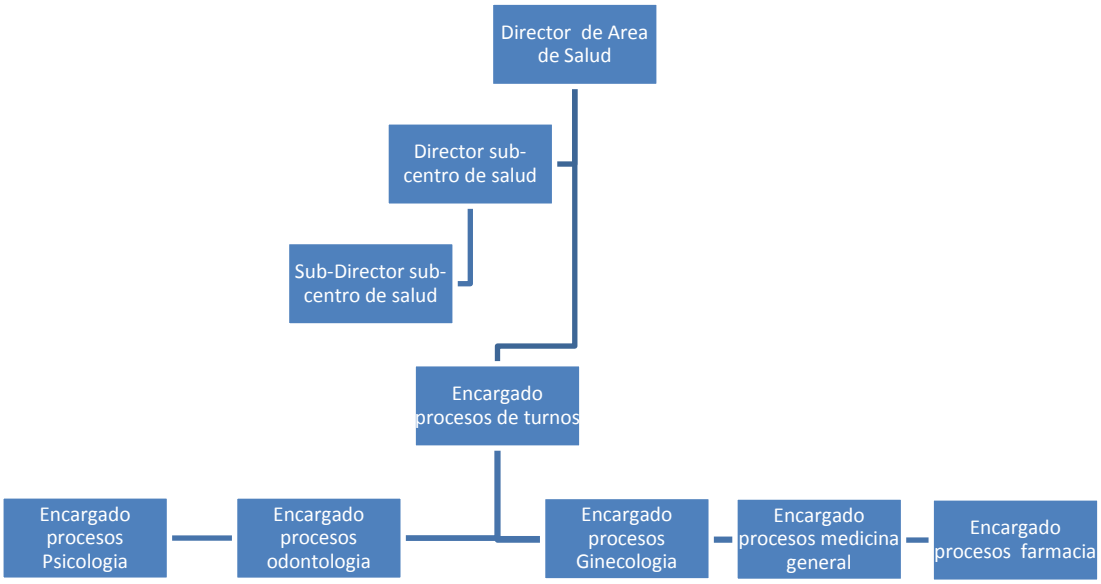


Figura I
Flujo de trabajo M.S.P.

"Un proceso asistencial lo podemos definir como el conjunto de fases o actividades sucesivas que tienden a conservar o mejorar la situación clínica de un paciente.

Pueden existir tantos procesos como quiera definir la empresa, por ejemplo, Procesos Farmacéuticos, Pediátricos, Neurológicos, Ginecológicos, Traumatológicos, etc., donde en cada uno se atiende cada patología relacionada, es decir, en los procesos Ginecológicos se tratarán las patologías ginecológicas, en los procesos neurológicos se tratarán las patologías neurológicas y así sucesivamente."[1]

4. Justificación

El problema observado por el equipo de salud desde hace ya varios años ha sido que los usuarios acudan a la madrugada a la unidad médica para poder alcanzar un turno en medicina general, obstetricia y odontología especialmente, tomando en cuenta la situación de riesgo de ser asaltado y de enfermar por el peligro y clima existente, sumado a esto la espera para ser atendidos.

Con estos antecedentes se desarrollaron algunas propuestas de mejoramiento y solución al problema, entre ellas la decisión de entregar turnos mediante vía telefónica con lo que se logrará que los usuarios ya no madruguen.

Con la preocupación por resolver este problema, el director del Área de Salud y la directora del SCS de Salud Carapungo tomaron la decisión de dar citas según horario y profesional que el usuario escoja, cuya evaluación se hará posteriormente.

Por estas razones hemos planteado la siguiente propuesta que creemos disminuirá el tiempo de espera del usuario, permitiendo que éste escoja el horario más adecuado y así pueda utilizar su tiempo en otras actividades, además se utilizará el tiempo de espera en dar a conocer temas sobre promoción, prevención y cuidado de la salud de los usuarios, la familia y la comunidad. Y hacer de la espera una estancia agradable, en un ambiente cómodo, limpio que cumpla con las expectativas del usuario.

[1] FUENTE: http://www.gestiopolis.com/canales7/fin/costos-en-las-empresas-de-salud.htm

5. Objetivos

a) Objetivo General:

Desarrollar e implementar el sistema que permita ofrecer una mejor atención al paciente, reduciendo los tiempos de espera, brindándole confort y actividades de promoción y prevención de la salud individual y familiar.

b) Objetivos Específicos:

- Mejorar la disponibilidad de los turnos para pacientes con patologías prevalentes disminuyendo el tiempo de espera.
- Facilitar la obtención de turnos para los estudios complementarios.
- Disminuir el tiempo de espera para la atención.
- Administrar doctores, horarios, áreas y pacientes.

6. Alcance

El proyecto concluirá al implementar el Sistema de Gestión de Turnos para el Sub-centro de Salud Carapungo, cabe recalcar que el sistema no integrará módulos como el historial clínico de cada paciente, de acuerdo a la aceptación del Director del Área de salud, además se entregará un documento que se emitirá por parte del Sub-centro de salud que detalle el trabajo realizado, adicionando toda la documentación que respalde el trabajo.

7. Metodología y Técnicas

Para el desarrollo e implementación del Sistema de Turnos Programados para el Sub-centro de salud Carapungo utilizaremos programación extrema (XP) como metodología de desarrollo durante todo el proceso, lo que implicará el lanzamiento de

varias versiones de prueba para el usuario. Además se recurrirá a fases de la Ingeniería de Software como análisis de factibilidad, análisis de requerimientos, planeación, diseño, desarrollo, pruebas e implantación. Durante el transcurso de las fases se procederá a realizar los distintos diagramas para modelar el sistema como por ejemplo BPMN para diseñar los distintos procesos del sistema.

El sistema será orientado hacia web, por lo que se lo implementará con leguaje PHP dado a su gran compatibilidad con la mayoría de plataformas y velocidad de ejecución, y la base de datos con MySql por ser de licencia gratuita y con grandes ventajas como fiabilidad y rapidez.

8. Hipótesis o supuestos

Teniendo en cuenta que vamos a utilizar la metodología Extreme Programming, se podrá ver las necesidades que se van presentando por parte del usuario y podremos acoplarlas al proyecto ya que se tendrá un buen diseño del sistema que será susceptible a cambios futuros.

Teniendo una buena comunicación con los beneficiarios del sistema podremos obtener más información de cómo se maneja los procesos en el sub-centro de salud para poder integrar el sistema a cada área médica y ayudar a agilitar su trabajo.

Toda la documentación y sistematización del programa será entregada de acuerdo al cronograma establecido con eso no tendremos inconvenientes en la realización y posterior entrega del proyecto.

9. Precondiciones

Conseguir ayuda por parte del personal para que nos brinde las facilidades para la obtención de la información requerida para la realización del sistema, mediante documentación necesitaría, entrevistas personales o reuniones grupales.

Tener claro todos los requerimientos y necesidades propuestos por el sub-centro de salud para el sistema y así evitar contratiempos con el cliente.

Obtener datos preliminares de las condiciones actuales del Sub-centro de salud para la posterior comparación con los datos a futuro y ver los beneficios que se obtendrá con el sistema.

El compromiso por parte de nuestro grupo de trabajo (Richard Siza, Raúl Landázuri) de proporcionar un producto de calidad y libre de errores para que el cliente se encuentre satisfecho del trabajo realizado.

10. Indicadores

Reducir o eliminar el tiempo de espera al paciente para conseguir un turno.

Tener un registro de asistencias de pacientes, esto ayudará a que las personas más cumplidas tengan preferencia al conseguir un turno.

Agilitar la búsqueda de los datos de un paciente.

Tener un producto de excelente calidad, ya que el método a utilizar es el de programación extrema y con esto se consigue tener más contacto con el cliente, el cual podrá interactuar con el programa y poder ver los pro y los contra del mismo, y así obtener un producto con cero errores y agradable para el usuario.

11. Fuentes de verificación

Nuestro proyecto será evaluado por parte del usuario cada vez que se tenga un avance y con esto se podrá verificar el correcto funcionamiento del mismo.

12. Sostenibilidad

El sistema se podrá mantener o mejorar en cuanto a su funcionalidad ya que lo realizaremos con el método de programación extrema con el cual podremos efectuar cambios que sean necesarios.

13. Medios o insumos

Para poder ejecutar las actividades previstas en nuestro proyecto necesitaremos de tiempo para realizar las reuniones con la parte beneficiaria por lo tanto se requiere de un espacio físico con un computador provisto con lo necesario para la posterior exposición del sistema.

Personas involucradas en cada actividad

a. Dirección de Área de Salud

Dr. Galo Barragán

b. Dirección sub-centro de salud

Dra. María Teresa Altamirano

Dr. Antonio Rodríguez (E)

c. Personal de cada especialidad del sub-centro de salud

Ginecología

Dra. Marisol Duque

Dra. Patricia Zaruma

Medicina general

Dra. María Teresa Altamirano

Dra. Berenice Arrobo

Pediatría

Dra. Fernanda Jiménez

Odontología

Dr. Fernando Mancero

Farmacia

Eco. Eduardo Solís

Estadística

Srta. Pamela Caicedo

Enfermería

En. Paola Mafla

d. Directores y revisores de tesis

Director: Ing. Jorge Alarcón

Revisores: Ing. Alfredo Calderón

Ing. Eddy Sánchez

14. Costos y presupuestos

Los costos que implican el sistema son mínimos ya que el sub-centro de salud se encuentra dentro del Distrito Metropolitano de Quito y nos resulta fácil llegar a él.

15. Matriz de marco lógico

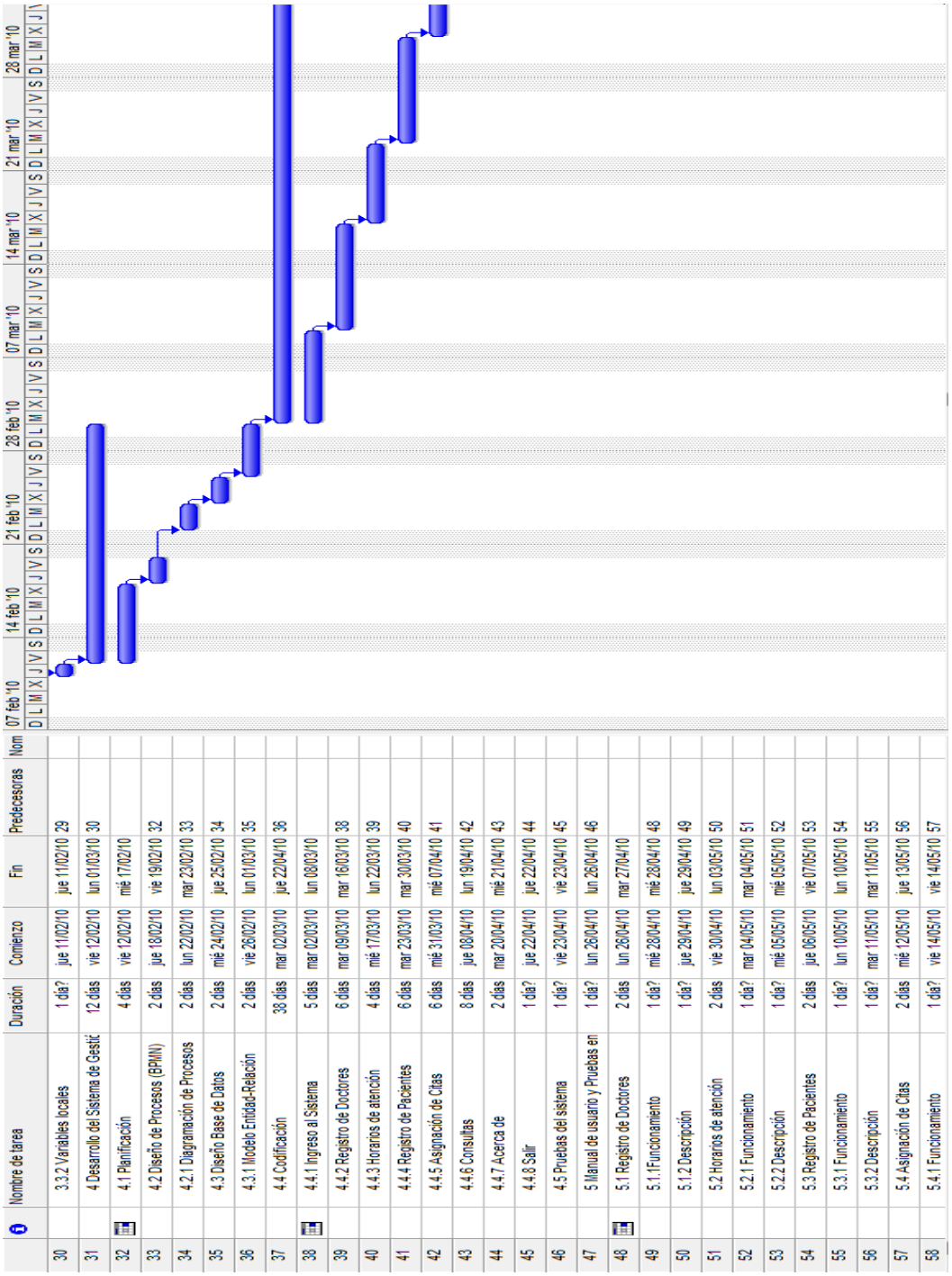
	DESCRIPCIÓN	INDICADORES	FUENTES DE VERIFICACIÓN	SUPUESTOS
Objetivo de desarrollo	o Obtener los requerimientos del sistema o Llevar un cronograma de actividades o Implementar el programa	o Obtener información por parte de los beneficiarios o Realizar las tareas en el tiempo establecido o Usar la metodología planteada	o Entrevistas con los trabajadores o Cronograma o Periódicas entrevistas con el cliente	o Facilidades para la obtención de los requerimientos o Seguir el cronograma de actividades o Tiempo del cliente.
Objetivo del proyecto	o Diseño, desarrollo e implementación del sistema de gestión de turnos programados para el Sub-centro de Salud	o Realizar una planificación adecuada	o Realizar diagramas de flujo	o Seguir la planificación descrita en el proyecto
Resultados esperados	o Generar un sistema que	o Obtener todos los	o Plasmar en los diagramas los	o Contar con la

	vaya de acuerdo con las necesidades del cliente o Facilitar el trabajo de la persona responsable en impartir los turnos	requerimientos del sistema o Realizando un sistema amigable y de fácil manejo	requerimientos descritos o Realizar una interfaz agradable al usuario	información correspondiente o Capacitación al usuario

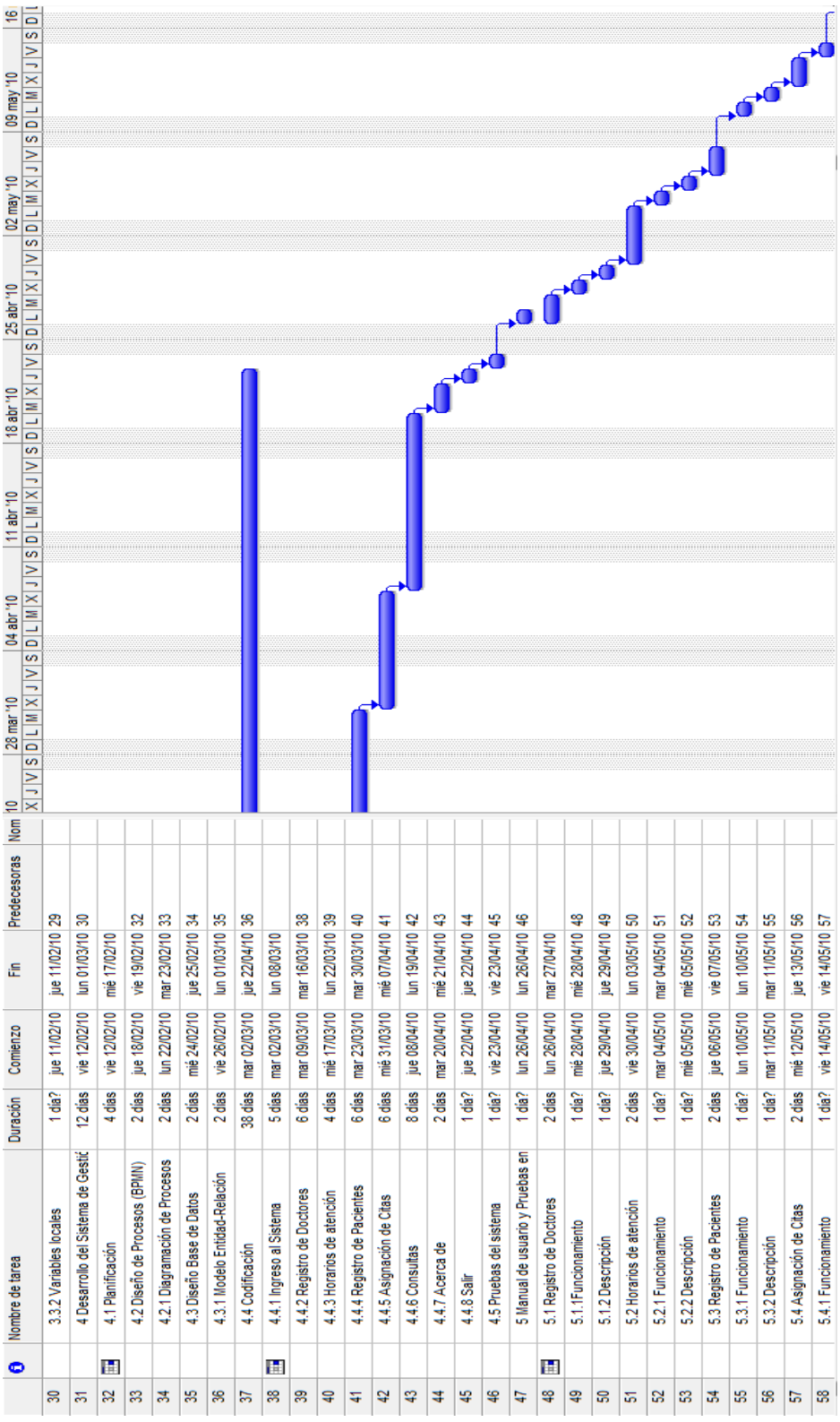
Actividades		**Insumos**	**Costos**	**Precondiciones**
Diseño		Diagramas de flujo, herramientas de diseño	5 $	obtención de información
Desarrollo		Metodología, Herramientas	5$	Conocimiento de las diferentes herramientas
Implementación		Software, Tecnología	10 $	Investigación de nuevas tecnologías

Tabla #1
Matriz de Marco Lógico.

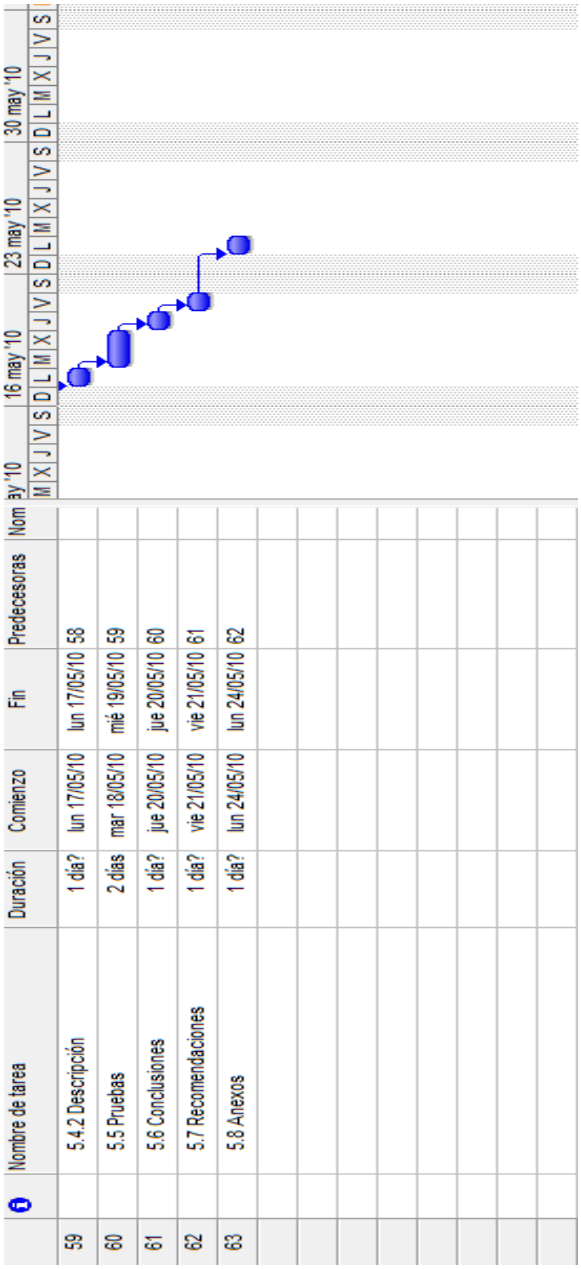
16. Cronograma de actividades



	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nom
30	3.3.2 Variables locales	1 día?	jue 11/02/10	jue 11/02/10	29	
31	4 Desarrollo del Sistema de Gestió	12 días	vie 12/02/10	lun 01/03/10	30	
32	4.1 Planificación	4 días	vie 12/02/10	mié 17/02/10		
33	4.2 Diseño de Procesos (BPMN)	2 días	jue 18/02/10	vie 19/02/10	32	
34	4.2.1 Diagramación de Procesos	2 días	lun 22/02/10	mar 23/02/10	33	
35	4.3 Diseño Base de Datos	2 días	mié 24/02/10	jue 25/02/10	34	
36	4.3.1 Modelo Entidad-Relación	2 días	vie 26/02/10	lun 01/03/10	35	
37	4.4 Codificación	38 días	mar 02/03/10	jue 22/04/10	36	
38	4.4.1 Ingreso al Sistema	5 días	mar 02/03/10	lun 08/03/10		
39	4.4.2 Registro de Doctores	6 días	mar 09/03/10	mar 16/03/10	38	
40	4.4.3 Horarios de atención	4 días	mié 17/03/10	lun 22/03/10	39	
41	4.4.4 Registro de Pacientes	6 días	mar 23/03/10	mar 30/03/10	40	
42	4.4.5 Asignación de Citas	6 días	mié 31/03/10	mié 07/04/10	41	
43	4.4.6 Consultas	8 días	jue 08/04/10	lun 19/04/10	42	
44	4.4.7 Acerca de	2 días	mar 20/04/10	mié 21/04/10	43	
45	4.4.8 Salir	1 día?	jue 22/04/10	jue 22/04/10	44	
46	4.5 Pruebas del sistema	1 día?	vie 23/04/10	vie 23/04/10	45	
47	5 Manual de usuario y Pruebas en	1 día?	lun 26/04/10	lun 26/04/10	46	
48	5.1 Registro de Doctores	2 días	lun 26/04/10	mar 27/04/10		
49	5.1.1Funcionamiento	1 día?	mié 28/04/10	mié 28/04/10	48	
50	5.1.2 Descripción	1 día?	jue 29/04/10	jue 29/04/10	49	
51	5.2 Horarios de atención	2 días	vie 30/04/10	lun 03/05/10	50	
52	5.2.1 Funcionamiento	1 día?	mar 04/05/10	mar 04/05/10	51	
53	5.2.2 Descripción	1 día?	mié 05/05/10	mié 05/05/10	52	
54	5.3 Registro de Pacientes	2 días	jue 06/05/10	vie 07/05/10	53	
55	5.3.1 Funcionamiento	1 día?	lun 10/05/10	lun 10/05/10	54	
56	5.3.2 Descripción	1 día?	mar 11/05/10	mar 11/05/10	55	
57	5.4 Asignación de Citas	2 días	mié 12/05/10	jue 13/05/10	56	
58	5.4.1 Funcionamiento	1 día?	vie 14/05/10	vie 14/05/10	57	



	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nom
30	3.3.2 Variables locales	1 día?	jue 11/02/10	jue 11/02/10	29	
31	4 Desarrollo del Sistema de Gestió	12 días	vie 12/02/10	lun 01/03/10	30	
32	4.1 Planificación	4 días	vie 12/02/10	mié 17/02/10		
33	4.2 Diseño de Procesos (BPMN)	2 días	jue 18/02/10	vie 19/02/10	32	
34	4.2.1 Diagramación de Procesos	2 días	lun 22/02/10	mar 23/02/10	33	
35	4.3 Diseño Base de Datos	2 días	mié 24/02/10	jue 25/02/10	34	
36	4.3.1 Modelo Entidad-Relación	2 días	vie 26/02/10	lun 01/03/10	35	
37	4.4 Codificación	38 días	mar 02/03/10	jue 22/04/10	36	
38	4.4.1 Ingreso al Sistema	5 días	mar 02/03/10	lun 08/03/10		
39	4.4.2 Registro de Doctores	6 días	mar 09/03/10	mar 16/03/10	38	
40	4.4.3 Horarios de atención	4 días	mié 17/03/10	lun 22/03/10	39	
41	4.4.4 Registro de Pacientes	6 días	mar 23/03/10	mar 30/03/10	40	
42	4.4.5 Asignación de Citas	6 días	mié 31/03/10	mié 07/04/10	41	
43	4.4.6 Consultas	8 días	jue 08/04/10	lun 19/04/10	42	
44	4.4.7 Acerca de	2 días	mar 20/04/10	mié 21/04/10	43	
45	4.4.8 Salir	1 día?	jue 22/04/10	jue 22/04/10	44	
46	4.5 Pruebas del sistema	1 día?	vie 23/04/10	vie 23/04/10	45	
47	5 Manual de usuario y Pruebas en	1 día?	lun 26/04/10	lun 26/04/10	46	
48	5.1 Registro de Doctores	2 días	lun 26/04/10	mar 27/04/10		
49	5.1.1Funcionamiento	1 día?	mié 28/04/10	mié 28/04/10	48	
50	5.1.2 Descripción	1 día?	jue 29/04/10	jue 29/04/10	49	
51	5.2 Horarios de atención	2 días	vie 30/04/10	lun 03/05/10	50	
52	5.2.1 Funcionamiento	1 día?	mar 04/05/10	mar 04/05/10	51	
53	5.2.2 Descripción	1 día?	mié 05/05/10	mié 05/05/10	52	
54	5.3 Registro de Pacientes	2 días	jue 06/05/10	vie 07/05/10	53	
55	5.3.1 Funcionamiento	1 día?	lun 10/05/10	lun 10/05/10	54	
56	5.3.2 Descripción	1 día?	mar 11/05/10	mar 11/05/10	55	
57	5.4 Asignación de Citas	2 días	mié 12/05/10	jue 13/05/10	56	
58	5.4.1 Funcionamiento	1 día?	vie 14/05/10	vie 14/05/10	57	



	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nom
59	5.4.2 Descripción	1 día?	lun 17/05/10	lun 17/05/10	58	
60	5.5 Pruebas	2 días	mar 18/05/10	mié 19/05/10	59	
61	5.6 Conclusiones	1 día?	jue 20/05/10	jue 20/05/10	60	
62	5.7 Recomendaciones	1 día?	vie 21/05/10	vie 21/05/10	61	
63	5.8 Anexos	1 día?	lun 24/05/10	lun 24/05/10	62	

CAPÍTULO I: Análisis del sistema de gestión de turnos programados

1.1 Introducción

El sistema se realizó en base a la necesidad descrita por el director del área de salud de Calderón el Dr. Galo Barragán quién manifestó que existe una gran afluencia de gente para tomar un turno para consulta externa en el sub-centro de salud de Carapungo, que se debe a la gran población que hay en ese sector, por tal razón el sistema facilitó la obtención de los turnos.

A la fecha de inicio de la disertación, el sub-centro de salud no dispone de una herramienta para gestión de turnos, lo cual es un inconveniente tanto para la administración del sub centro de salud como para los pacientes. En una visita al centro médico nos pudimos dar cuenta que la fila que hacen los pacientes es muy grande y no todos pueden ser atendidos, al igual que la persona responsable de asignar los turnos tiene un arduo trabajo con cada persona que solicita uno. Debido a estas razones el sub-centro de salud solicitó la realización del sistema de gestión de turnos para agilitar y brindar una mejor atención a los pacientes.

En acción a los requerimientos expuestos en el centro de salud el día viernes 18 de diciembre de 2009, nosotros (Richard Siza y Raúl Landázuri) utilizamos la metodología de Programación Extrema para la realización del sistema con esto interactuamos más con los beneficiarios, ya que utilizaron el producto antes que se encuentre finalizado, por lo que ésta metodología nos permitió realizar los cambios respectivos según la necesidad del cliente, con lo cual obtuvimos un producto de calidad y la satisfacción del cliente.

1.2 **Descripción de la unidad operativa**

El sub-centro de salud de Carapungo es una unidad médica que fue construida hace 15 años y desde entonces ha brindado atención médica ambulatoria. El sub-centro de salud ha ido modificando su infraestructura física por lo cual ha aumentado sus especialidades y actualmente posee las siguientes prestaciones:

- Medicina Familiar
- Medicina General
- Ginecología
- Obstetricia
- Odontopediatría
- Odontología
- Enfermería
- Vacunas
- Laboratorio Clínico
- Farmacia

1.3 **Herramientas utilizadas en el proyecto**

1.3.1. **Diseñar el sistema**

1.3.1.1. **CASE**

Las herramientas CASE[1] (Computer Aided Software Engineering), son aplicaciones que ayudan a aumentar la productividad en el desarrollo de software bajando el costo de la misma ya que se reduce el tiempo y la inversión. Esta herramienta será de gran uso 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 costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

[1] Diversas aplicaciones de software utilizadas para el desarrollo de software, reduciendo tiempo y dinero empleado.

Objetivos de la Herramienta CASE:

- Mejorar la productividad en el desarrollo y mantenimiento del software.
- Aumentar la calidad del software.
- Reducir el tiempo y coste de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

Por estos objetivos que presentan las herramientas CASE nos vemos en la necesitad de emplearlas en nuestro sistema para que salga un producto con calidad, eficiente y eficaz.

1.3.1.2. **BPMN**[2]

La BPMN permite expresar gráficamente procesos de negocio o flujos de trabajo de una forma comprensible para todos los usuarios de una organización.

[2] Notación gráfica que permite el modelamiento de procesos de negocio

Si consideramos que la gestión se realiza a través de todo el ciclo de vida de BPMN, es indudable que el modelamiento tiene un papel preponderante. Asimismo, un diagrama en BPMN puede ser transformado en un código ejecutable automáticamente, sin la necesidad de programación. De esta forma, el analista de negocios puede definir, diseñar y generar una solución a sus procesos.

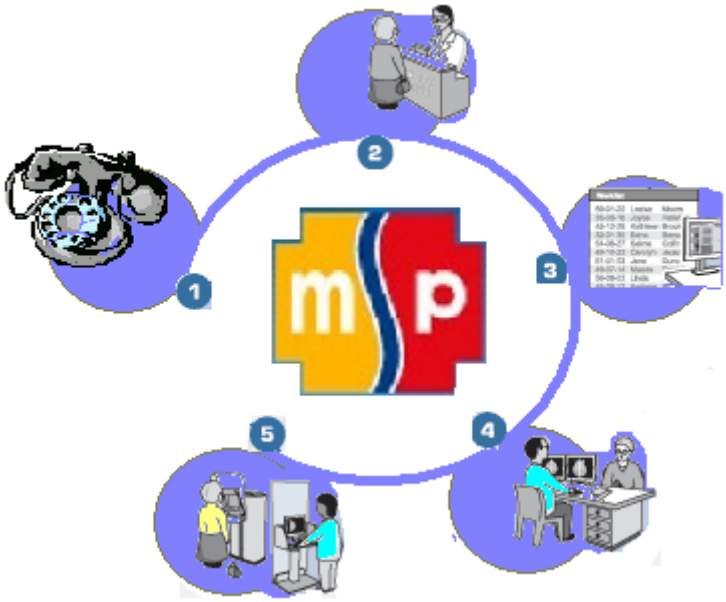
Para realiza el diseño del sistema de gestión de turnos programados, ésta metodología ofrece un único diagrama con hiperlinks, el cual contempla desde la concepción general hasta la generación del código ejecutable de los procesos, pasando por diagramas de detalle, tratamiento de excepciones y otros.

1.3.1.3. **WORKFLOW**[3] (flujo de trabajo)

Se refiere al flujo de trabajo a seguir para la consecución de una tarea o trabajo predeterminado. El workflow general de una empresa presenta las actividades a realizarse así como los tiempos y organización de las mismas.

Se trata de la parte de un proceso de negocio que admite la automatización de procedimientos o flujos de trabajo donde documentos, información y/o tareas son pasados de un participante a otro, en un camino gobernado por reglas o procedimientos

[3] Estudio de aspectos operaciones de una actividad de trabajo.



Autores: Landázuri Raúl, Siza Richard.
Fecha: 04 – 04 – 2010
Descripción: Flujo de actividades que realiza el S.C.S Carapungo para la entrega de citas médicas.

Figura #2
Flujo de trabajo M.S.P.

1.3.2. **Desarrollo del sistema**

1.3.2.1 PHP[4]

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación, ya que está diseñado especialmente para desarrollo web y puede ser incrustado dentro de código HTML[5]. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, lo que implica mayor popularidad y adaptación al medio.

[4] Lenguaje de programación hecho originalmente para la creación de páginas web.
[5] HyperText Markup Language, lenguaje predominante para la creación de páginas web.

Por ser un lenguaje gratuito no depende alguna compañía para su soporte o ayuda, sin dejar a un lado que por el hecho de que funciona bajo "casi" todos los sistemas operativos, los mismos usuarios han creado varios manuales y tutoriales claramente explicando su funcionamiento. Por estas razones escogimos este lenguaje de programación para desarrollar el sistema de gestión de turnos programados para el SCS Carapungo.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 04 – 04 – 2010
Descripción: Representación gráfica de PHP.
Fuente: http://www.adrformacion.com/curso/php/leccion1/diagrama-php.gif

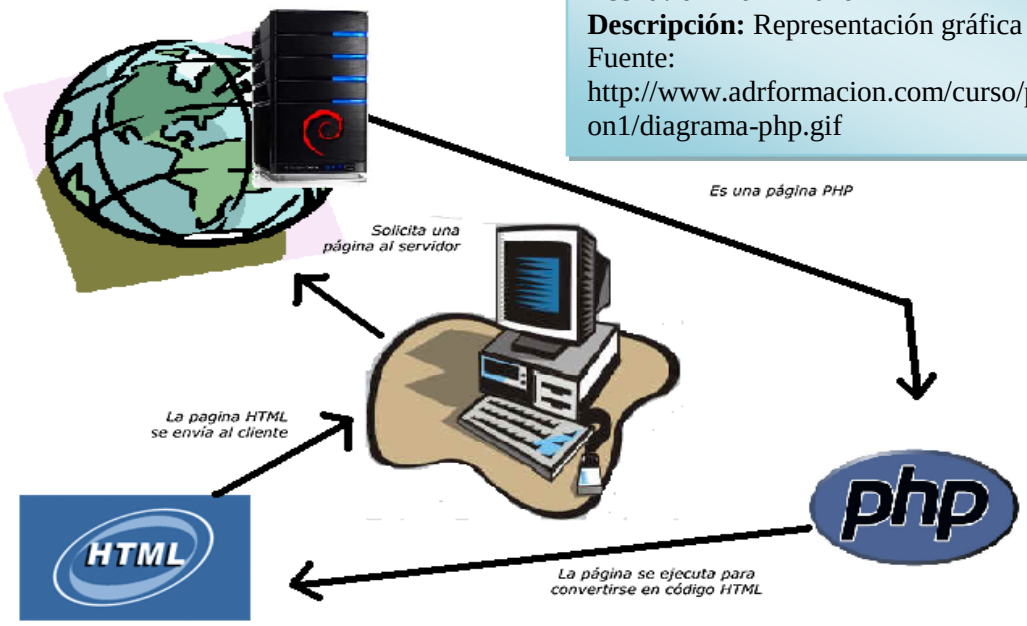


Figura #3
Representación gráfica de PHP

1.3.2.2 XAMPP

XAMPP es una forma fácil de instalar la distribución Apache que contiene principalmente la base de datos MySQL[6], el servidor Web Apache y los

[6] MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

intérpretes para lenguajes de script: PHP[7] y Perl. El nombre proviene del acrónimo de X (para cualquiera de los diferentes sistemas operativos), Apache, MySQL, PHP, Perl. El programa está liberado bajo la licencia GNU[8] y actúa como un servidor Web libre, fácil de usar y capaz de interpretar páginas dinámicas. Existen versiones para Linux**,** Windows**,** MacOS X **y** Solaris**,** cuyos paquetes difieren según la disponibilidad de los diversos programas en cada plataforma.

Decidimos utilizar este programa en la implementación de nuestro sistema, ya que, tiene integrado tanto la base de datos MySQL como el servidor APACHE, necesarios para el funcionamiento del sistema.

Autores: Landázuri Raúl, Siza Richard.
Fecha: 04 – 04 – 2010
Descripción: Panel de control de XAMPP.

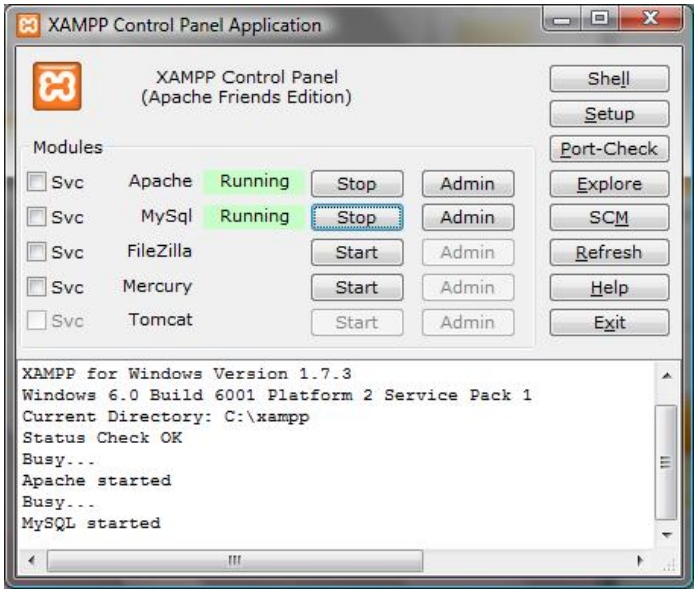


Figura #4
Interfaz gráfica de XAMPP.

[7] Es un lenguaje de propósito general utilizado para la administración de sistemas, desarrollo web, programación en red, desarrollo de GUI, entre otras.
[8] Licencia Pública General.

1.3.2.3 DREAM WEAVER

Es la aplicación de este tipo más usada en el sector de diseño y programación web por sus excelentes ayudas y aplicaciones. Posee, como toda la línea Macromedia/Adobe, excelentes funcionalidades e integración con otras herramientas como Adobe Flash[9].

La aplicación permite crear sitios de forma totalmente gráfica, y dispone de funciones para acceder al código HTML generado. Permite la conexión a un servidor, a base de datos, soporte para programación en ASP[10], PHP, Java script, cliente FTP integrado, etc.

Decidimos utilizar este IDE de desarrollo debido a que obtuvimos licencias educativas del programa, y porque está enfocado netamente en la implementación hacia la web.

Autores: Landázuri Raúl, Siza Richard.
Fecha: 04 – 04 – 2010
Descripción: Interfaz gráfica de Dreamweaver

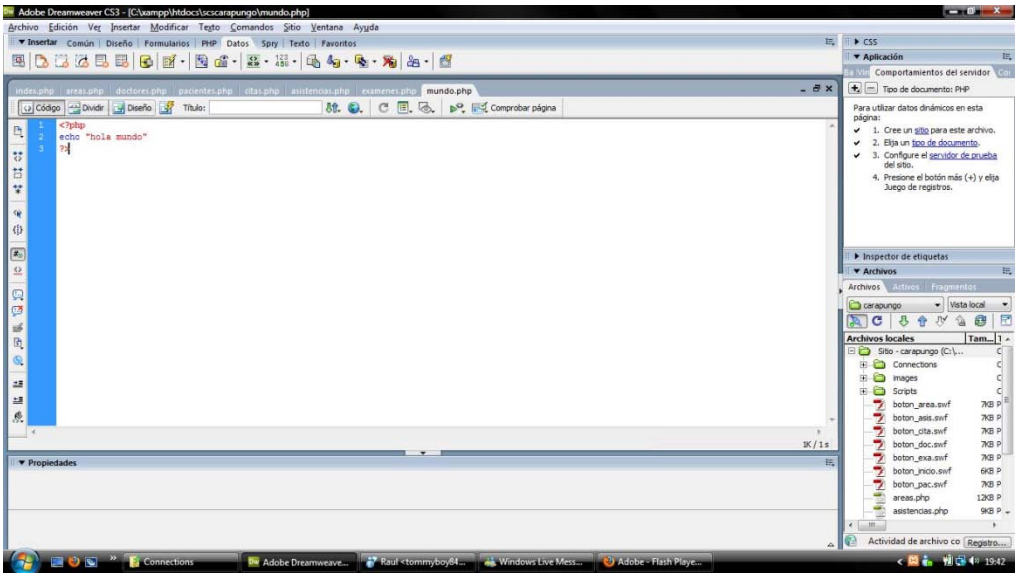


Figura #5
Interfaz gráfica de Dreamweaver.

[9] Es una aplicación en forma de estudio de animación que trabaja sobre fotogramas.
[10] Es una tecnología del lado servidor de Microsoft para páginas web generadas dinámicamente.

1.3.2.4 **PHP Designer**

Es una plataforma de programación y desarrollo. Es muy utilizada para trabajar con PHP, pero de igual forma se utiliza para otros lenguajes como HTML, XHTML, SQL, etc.

Php designer es una herramienta con una interfaz muy amigable, la misma que ofrece toda una serie de asistentes y diálogos integrados y nos ayuda a realizar nuestro trabajo.

Posee en esquema práctico con colores lo que nos ayuda a distinguir la sintaxis del código fuente, además cuenta con la opción de autocorrección y autocompletado lo cual le hace una plataforma muy completa.

Este programa tiene cualidades muy buenas como las ya antes mencionadas, pero decidimos no utilizarlo porque no teníamos licencia la licencia para poder utilizarlo, también era un IDE nuevo para nosotros, lo que implicaba pérdida de tiempo hasta poder acoplarnos a la nueva herramienta.

1.3.2.5 POWER DESIGNER

PowerDesigner es un conjunto de herramientas de modelado que combina distintas técnicas estándar:

- Modelado de aplicación a través de UML[11]
- Técnica de Modelado de Procesos Empresariales
- Técnicas tradicionales de modelado de base de datos.

PowerDesigner ofrece análisis de impacto de gran alcance, el diseño de la gestión del cambio del tiempo, y técnicas de gestión de metadatos a la empresa.

[11] Lenguaje Unificado de Modelado, Es un lenguaje que proporciona un medio gráfico de modelar varios componentes de un sistema de software

Nos permite hacer el diseño del sistema en distintos modelos, como el diagrama de casos de uso, diagrama de clases, modelo conceptual y físico de datos, etc.

Por la variedad de modelados que podemos realizar, además que permite el modelado de gran cantidad de bases de datos, la conexión a distintos motores de bases de datos, y lo más importante es que puede crear automáticamente el script de creación de la base de datos, escogimos esta herramienta para el diseño de nuestro sistema.

Autores: Landázuri Raúl, Siza Richard.
Fecha: 04 – 04 – 2010
Descripción: Interfaz gráfica de PowerDesigner.

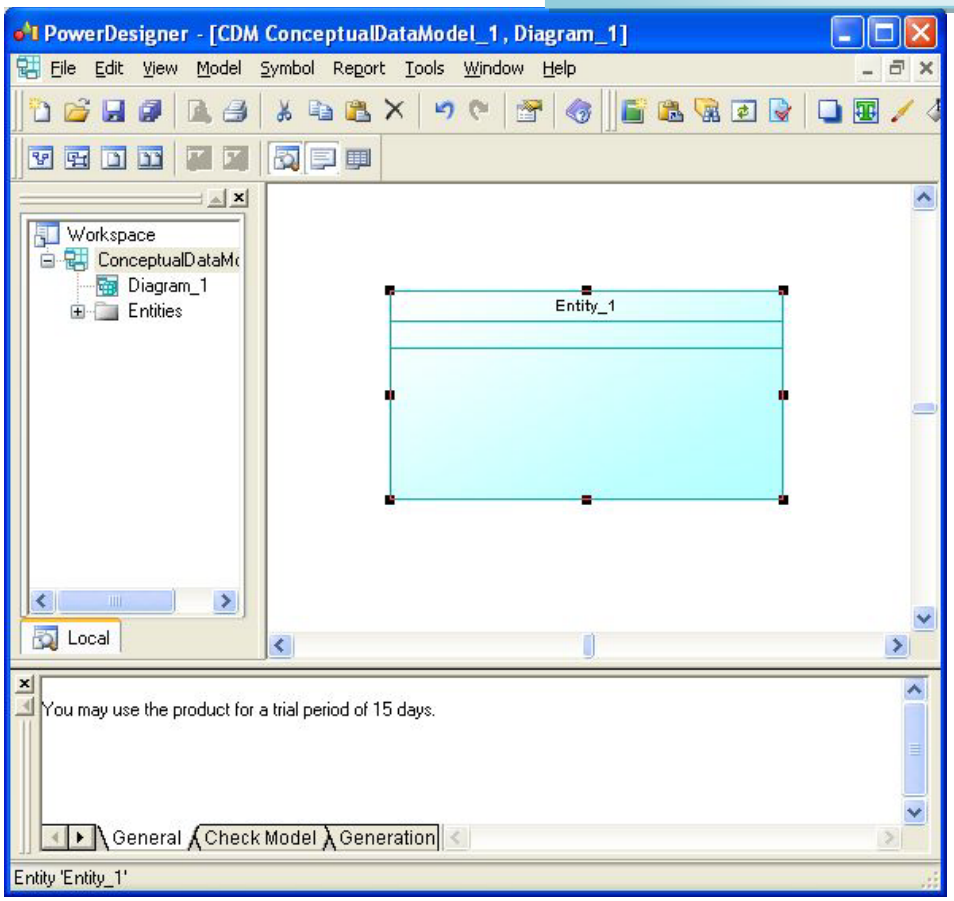


Figura #6
Interfaz gráfica de PowerDesigner.

1.3.2.6 APACHE

Apache es un programa de servidor HTTP Web de código abierto (open source). Fue desarrollado en los 90's. Su desarrollo comenzó en febrero de 1995 y actualmente es uno de los servidores web más utilizados en la red. Usualmente corre sobre UNIX, Linux, BSD y Windows. Es un poderoso paquete de servidor web con muchos módulos que se le pueden agregar y que se consiguen gratuitamente en el Internet. El nombre "Apache" es un acrónimo de "a patchy server" un servidor de remiendos, es decir un servidor construido con código preexistente y piezas y parches de código.

Por ser de código abierto al igual que el lenguaje de programación que escogimos anteriormente, decidimos utilizar este servidor para nuestro sistema.

1.3.2.7 Oracle

Es un sistema administrador de base de datos relacional o RDBMS (Relational Data Base Mangement System) desarrollado por Oracle Corporation. Es considerado como uno de los sistemas de base de datos más completos, incluyendo entre sus cualidades el soporte de transacciones o procesos, estabilidad, escalabilidad y lo más importe el soporte multiplataformas. Oracle se basa en una tecnología cliente/servidor muy apetecible a nivel mundial por sus reconocidas características. Este producto suele utilizar PL/SQL como lenguaje de programación para crear sus consultas y gestionar la base de datos.

Esta herramienta es mundialmente utilizada por grandes multinacionales por su gran potencia y elevados costos, lo que hace que otras empresas utilicen bases de datos como Access, MySql, SQL Server, etc.

Fecha: 04 – 04 – 2010
Descripción: Logo Oracle.
Fuente:http://www.sondeoeconomico.com/wp-content/uploads/2007/10/oracle.gif

Figura #7
Logo Oracle.

1.3.2.8 PostgreSQL

Sistema opensource gestor de base de datos relacional orientada a objetos, dirigido por una comunidad de desarrolladores denominada PGDG (PostgreSQL Global Development Group).

PostgreSQL permite el acceso simultáneo a la base de datos, es decir, se puede realizar procesos sobre la base mientras usuarios ingresen a la misma sin ningún tipo de bloqueo.

También posee tecnología cliente/servidor y otras características como integridad referencial, lenguajes procedurales, entre otros, lo que lo convierte en el sistema de base de datos de código abierto más avanzado del mercado.

Fecha: 04 – 04 – 2010
Descripción: Logo PostgreSQL.
Fuente:http://dattatecblog.com/wp-content/uploads/2009/05/postgresql.png

Figura #8
Logo PosterSQL.

CAPÍTULO II: Programación Extrema (XP)

2.1 Introducción

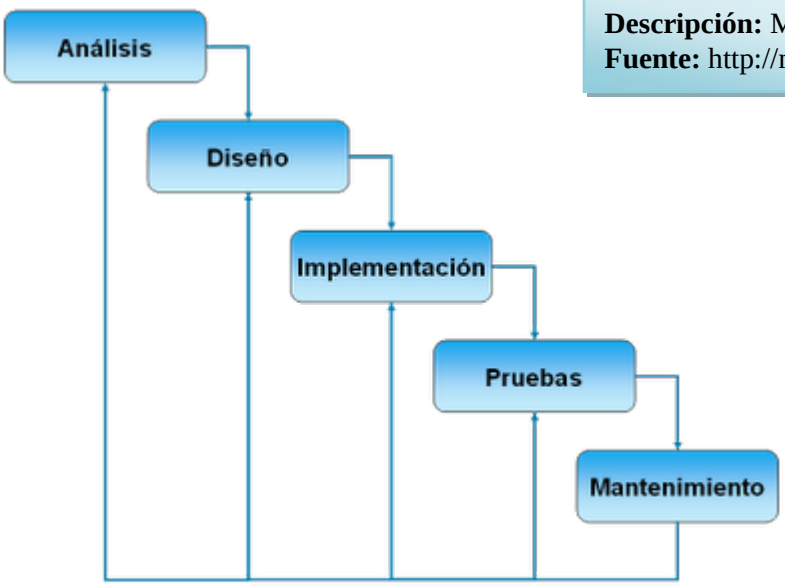
La ingeniería de software define métodos para el desarrollo de software, estos métodos son un conjunto de reglas definidas fácilmente enseñadas, repetidas y utilizadas por grupos de personas que cooperan entre sí para desarrollar software.

La ingeniería de software tiene varios modelos de desarrollo que nos pueden ayudar para la realización de cualquier tipo de software, de los cuales vamos a destacar los más utilizados:

2.1.1 Modelo en cascada:

El Modelo en cascada, es un enfoque metodológico el cual se encarga de ordenar las etapas de ciclo de vida del software, esto quiere decir que se debe esperar que finalice una etapa para poder iniciar con la siguiente.

En el siguiente grafico mostramos el diagrama del modelo en cascada:



Fecha: 05 – 03 – 2010
Descripción: Modelo en cascada
Fuente: http://modeloencascada.blogspot.com/

Figura #9
Modelo en cascada.

Este modelo se recomienda usar en proyectos pequeños ya que es un modelo en línea y cuando se detecta un error en una etapa avanzada del proyecto nos tocaría rehacer el trabajo y eso en un proyecto grande sería catastrófico.

2.1.1.1 Ventajas:

La principal ventaja es que se lleva un orden y no se mezclan las fases

Se utiliza en proyectos donde se especifiquen bien los requerimientos para así no tener problemas posteriormente.

2.1.1.2 Desventajas:

Es un proceso que no es muy ágil ya que debemos terminar cada etapa para poder iniciar con otra y esto causa retraso en la ejecución del proyecto.

Cuando existe un error en una etapa avanzada del proyecto se necesita regresar a etapas anteriores.

No se puede utilizar en proyectos grandes a menos que estén bien planteados los requerimientos.

2.1.2 Modelo espiral:

Este modelo tiene mejoras al modelo en cascada, ya que Las actividades no están fijadas a priori, sino que se las eligen en función del análisis de riesgo. Cada bucle o iteración de este modelo representa un conjunto de actividades las cuales las presentamos en el siguiente grafico:

Fecha: 05 – 03 – 2010
Descripción: Modelo en espiral
Fuente:http://boanerges7.galeon.com/Imagenes/ESPIRAL.JPG

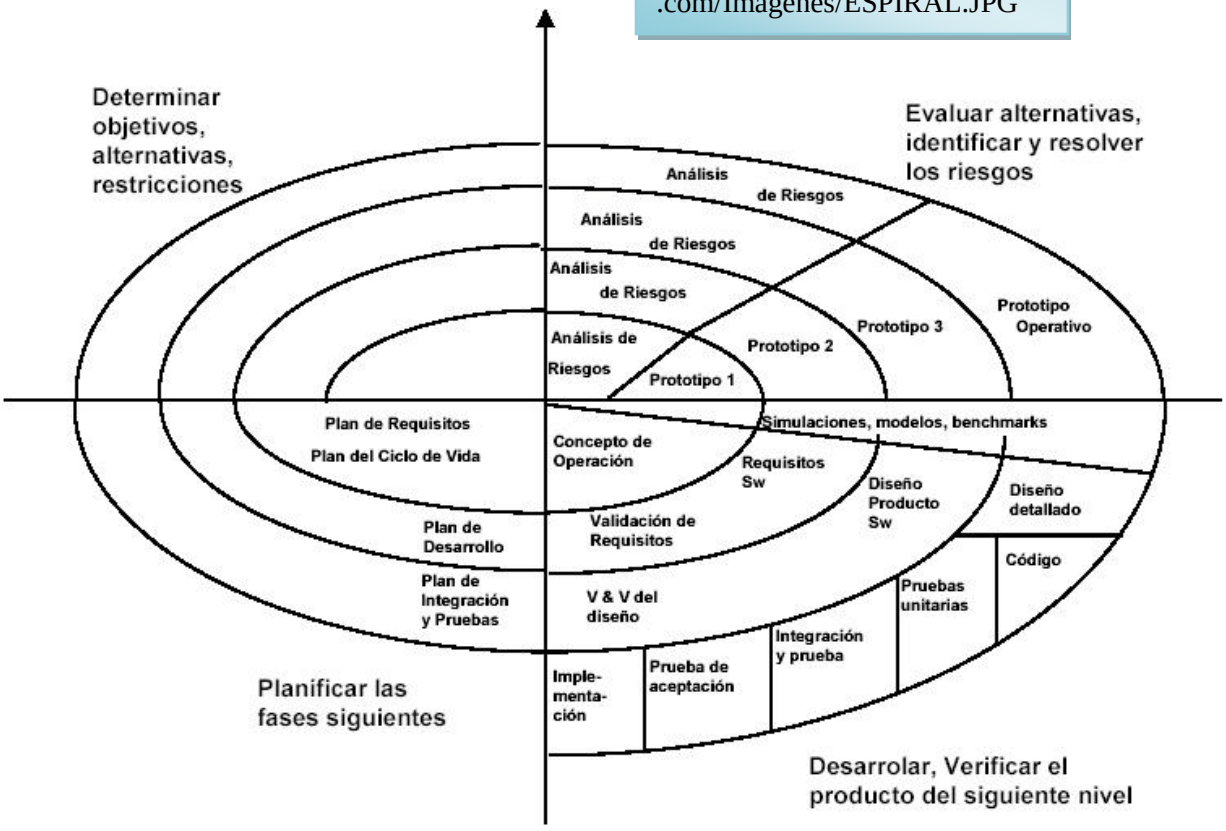


Figura #10
Modelo en espiral.

2.1.2.1 Ventajas:

Este modelo no es rígido con lo cual es susceptible a modificaciones y mejoras.

El análisis del riesgo se hace de forma clara con lo cual se reduce los riesgos del proyecto.

2.1.2.2 Desventajas:

Se requiere experiencia para el análisis de riesgos.

Este modelo es costoso y lleva tiempo planificar un proyecto.

2.1.3 Modelo de prototipos

Este modelo se enfoca en el desarrollo evolutivo, el modelo se caracteriza por ser rápido en la etapa de diseño, en el cual se construye un prototipo, el cual es revisado por el cliente y el usuario para realizar cambios en los objetivos y requerimientos si estos los requieren.

2.1.3.1 Ventajas:

La creación de prototipos ayuda al desarrollador de software y al cliente a entender de una mejor manera cuál será el resultado de la construcción del software.

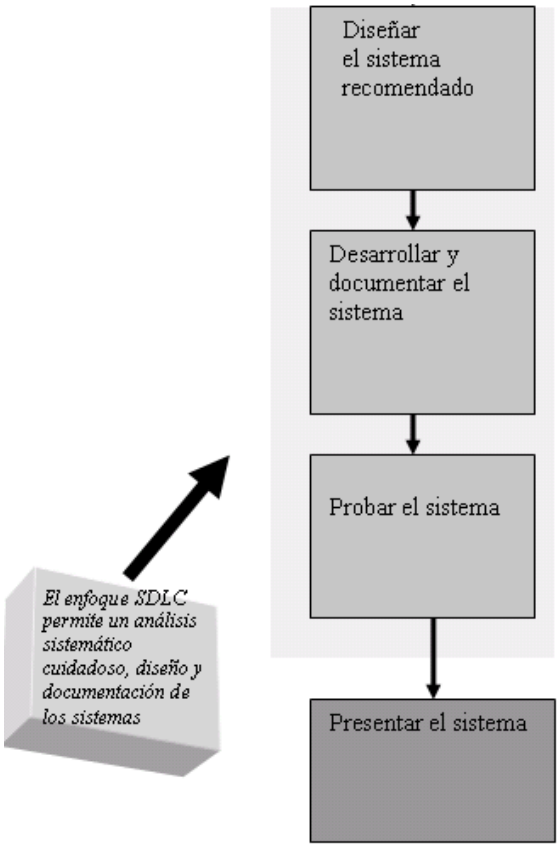
2.1.3.2 Desventajas:

El hecho que sea un método rápido provoca que el desarrollador tome algunas decisiones muy apresuradas por ejemplo, la elección del lenguaje de programación.

Otra desventaja es que al ver el prototipo el cliente se enfoca mas en el diseño del producto y deja de lado aspectos importantes, como la calidad y el mantenimiento a largo plazo del producto.

2.1.4 RAD[23].

Es un proceso de desarrollo de software que se caracteriza por su rapidez en el desarrollo de interfaces graficas de usuario, su usabilidad y su utilidad.



Fecha: 05 – 03 – 2010
Descripción: RAD.
Fuente: http://www.monografias.com/trabajos59/elaboracion-prototipos/Image24633.gif

Figura #11
RAD.

2.1.4.1 Ventajas:

Velocidad en el desarrollo.

Costos bajos debido a la rapidez de la construcción del sistema

[23] (Rapid Application Development) es un proceso de desarrollo de software.

2.1.4.2 Desventajas:

Pocas características del sistema.

Baja Escalabilidad[24].

2.1.5 Desarrollo por etapas:

Este modelo es similar al modelo por prototipos, ya q el cliente puede ver el software en las diferentes etapas del desarrollo, la única diferencia es que los requerimientos del sistema no están muy detallados por lo cual se van a ir presentando modificaciones a lo largo del proyecto.

2.1.5.1 Ventajas:

Es un método rápido y no es muy costoso.

2.1.5.2 Desventajas:

Al no tomar las especificaciones a detalle al inicio del proyecto se va a complicar la construcción del mismo, ya que no vamos a tener muy claro los requerimientos del cliente.

2.1.6 Desarrollo iterativo y creciente o iterativo o incremental:

Este modelo es una mejora al modelo en cascada. El desarrollo incremental e iterativo se basa en ir aprendiendo del desarrollo anterior e ir mejorando las versiones.

2.1.6.1 Ventajas:

El modelo tendrá pocos errores, ya que en cada iteración surgirá una mejora.

Es un proceso muy efectivo para que se tenga un producto de alta calidad.

2.1.6.2 Desventajas:

Es un proceso un poco demorado ya que cada iteración requiere de modificaciones.

[24] Capacidad de un software o de un hardware de crecer, adaptándose a nuevos requisitos conforme cambian las necesidades del negocio

Según Kent Beck[25] el modelo del proceso de desarrollo de sistemas que más adquiere popularidad y verdadera importancia es la programación extrema, ya que es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la retroalimentación. Programación extrema hace énfasis en la evaluación constante de los programas creando pruebas antes de la implementación, durante y cuando haya terminado.

2.2 **Que es Extreme Programming (XP)**

2.2.1 Historia de la programación extrema:

En el año de 1989, Cunningham creó un equipo con el cual trabajaba con las mismas prácticas y principios del XP. Pero Kent Beck fue el que implanto la propuesta dándole forma y nombre. Posteriormente Kent escribió el libro "Extreme Programming Explained"en el año 1999, el cual consolido la metodología de XP, ya que destaca la experiencia de trabajar con esta propuesta.

El primer proyecto que realizo Kent con programación extrema fue en Chrysler Corporation, la cual se le pidió hacer una aplicación. Y es aquí donde nace la programación extrema.

Otro proyecto en el que utilizaron programación extrema es el que realizaron los autores de la misma, el cual consistió en un sitio web "Portland Pattern Repository " [26] que sirve como un medio para ayudar a los programadores orientados a objetos publicar sus ideas y patrones de programación de la computadora mediante la presentación de estos. Además este portal Iba acompañado de un sitio web complementario, WikiWikiWeb[27].

[25] Autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999)).
[26] Patrones de diseño de programación informática
[27]"Wiki" es un sistema de composición, es un medio de discusión, es un repositorio, es un sistema de correo, es una herramienta de colaboración.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 05 – 03 – 2010
Descripción: Comparación Método ágil vs tradicional
Fuente: http://www.slideshare.net/edgarespinoza/programacion-extrema

COMPARACIÓN MÉTODO ÁGIL VS TRADICIONAL	
ÁGIL	**TRADICIONAL**
El cliente forma parte del desarrollo	El cliente mediante reuniones interactúa con los desarrolladores
Enfocado a proyectos pequeños, de corta duración y de entregas frecuentes	Orientado a proyectos de cualquier tamaño, pero suelen ser usados en proyectos grandes
El diseño se lo puede ir definiendo y mejorando de acuerdo a las necesidades del cliente	Se procura que el diseño se encuentre bien definido para que no exista problemas en el desarrollo del proyecto
Se espera que suceda cambios en el transcurso del proyecto	Se espera que no existan cambios que afecten en el desarrollo del proyecto
Pocos roles, más flexibles	Más roles, más específicos

Tabla #1
Comparación Método ágil vs tradicional.

Para realizar o desarrollar un proyecto debemos saber que es imposible conocer todas las necesidades del cliente, los requerimientos del sistema entre otras cosas que llevaría mucho tiempo planificar, por esta razón hemos decido trabajar con el método de programación extrema, el cual consiste en trabajar estrechamente con el cliente enseñándole versiones pequeñas del producto en el cual se está trabajando, por lo que se va a modificar o corregir para que el producto llene las expectativas del cliente. Además debemos tener presente que las metodologías ágiles están revolucionando la manera de producir software.

Para desarrollar un proyecto con XP debemos seguir algunos pasos:

El equipo de desarrollo debe trabajar junto al cliente y definir que se va hacer, analizar requerimientos y necesidades del proyecto, pero debemos saber que esta planificación es inexacta por lo que no sabemos qué cambios se pueden producir a lo largo de la realización del proyecto.

2.3 **Actividades de programación extrema**

Programación extrema se ha usado mucho últimamente, ya que es una metodología práctica para la creación y el ágil desarrollo de software. Entre sus principales actividades podemos mostrar las siguientes:

2.3.1 Codificación

Para todos los que hemos trabajado con programación extrema sabemos que la codificación tiene gran relevancia en el proceso de desarrollo del sistema porque aquí vamos a aplicar e implantar todo las exigencias expuestas por parte del cliente.

2.3.2 Pruebas

En programación extrema los desarrolladores no podemos estar seguros de que una aplicación trabaje a menos que uno la pruebe, estas pruebas se las realizara periódicamente alado del cliente para que el pueda ver su operatividad, rendimiento y de esta manera si realizamos todas las pruebas nosotros obtendremos un producto que tenga la aceptación del cliente y de muy buena calidad.

2.3.3 Escuchar

Al principio, institución o la entidad para la que se trabaje, por eso para comprender la funcionalidad que requiere el sistema es muy importante escuchar al cliente todos los requerimientos y aspectos importantes de la empresa y así obtener un producto que satisfaga las exigencias de los interesados.

2.3.4 Diseño

Para la mayoría de personas la programación extrema no necesita de diseño que solo basta con la codificación, pruebas y escuchar pero eso es porque han tenido un contacto muy rápido con esta metodología. Esto es porque en la programación extrema la fase de diseño es diferente a las demás metodologías. Además para esta fase se necesita tener habilidades para lograr hacer un diseño simple y que al usar refactorización el diseño se mantenga intacto.

2.4 **Ciclo de vida XP**

El ciclo de vida es la estructura del desarrollo de software, desde el nivel inicial al final.

El propósito del ciclo de vida es definir las diferentes fases intermedias que se requieren para el desarrollo de la aplicación, además este método sirve para detectar errores y posteriormente modificarlos y así llevar un software de buena calidad.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 05 – 03 – 2010
Descripción: Ciclo de vida de la programación extrema.
Fuente:http://www.monografias.com/trabajos51/programacion-extrema/Image3551.gif

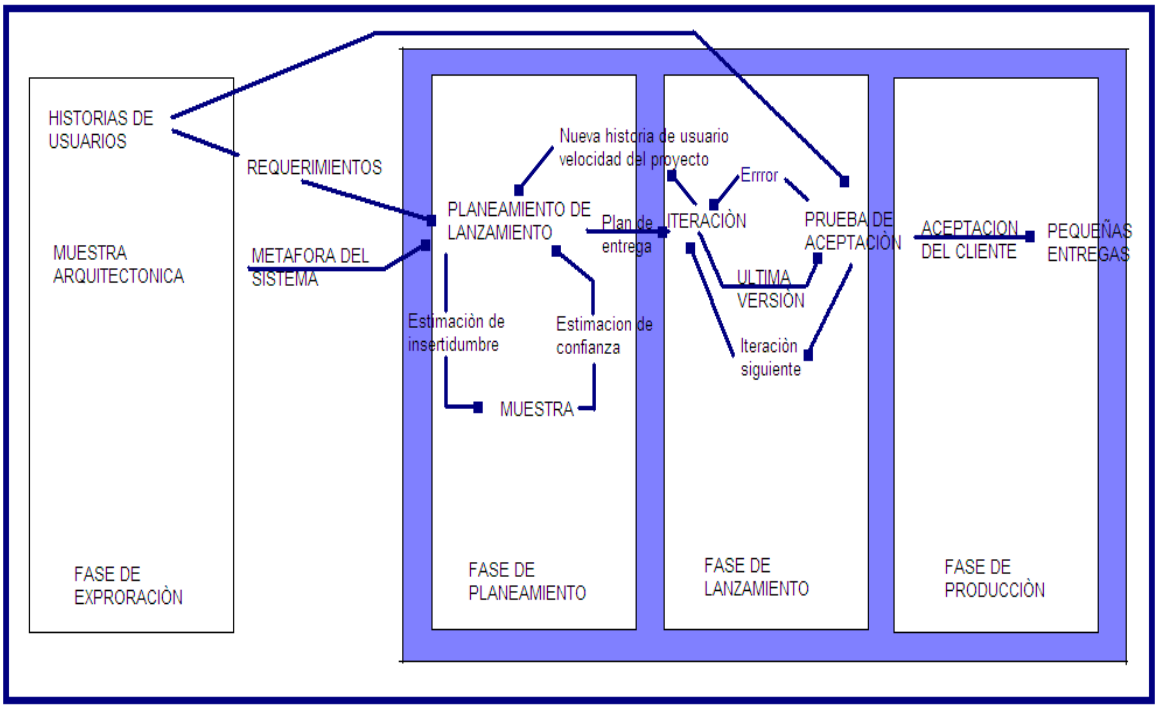


Figura #12
Ciclo de vida de XP.

2.4.1 Fase de exploración

Esta es la fase inicial donde los clientes exponen las características y necesidades del proyecto a realizar. Al mismo tiempo los desarrolladores nos podemos familiarizar con la última tecnología y herramientas disponibles.

La fase de exploración no requiere de mucho tiempo, todo depende del tamaño del proyecto.

En esta etapa debemos tener claro los siguientes puntos:

- Conocer de forma general las historias de usuario.
- Familiaridad con las herramientas y tecnologías por parte de los desarrolladores.
- Construcción de un prototipo del proyecto.

2.4.2 Fase de planeamiento

Luego de pasar la fase de exploración procedemos a realizar el cronograma del proyecto. En esta fase vamos a realizar un plan de lanzamiento el cual consiste en una planificación donde los desarrolladores y clientes establecen los tiempos de implementación y entrega, la prioridad con la que se implementará cada versión del programa.

Al finalizar el plan de lanzamiento debemos tener claro ciertos puntos:

- Los objetivos del proyecto.
- El tiempo que se tardará la entrega de cada versión y entrega del proyecto en su totalidad.
- El número de participantes del proyecto.
- La forma de evaluar el trabajo realizado.

2.4.3 Fase de producción

Esta fase requiere de comprobar y probar cada versión que se entregará al cliente. Además es importante que cuando se produzcan cambios se debe tomar la decisión de incluir o no en el lanzamiento actual caso contrario documentar esta información para poder ponerla en práctica en la siguiente fase.

En esta fase podemos destacar los siguientes puntos:

- Antes de cada entrega del sistema al cliente se debe una revisión extra por parte de los desarrolladores.
- Si se encuentran nuevos cambios tomar la decisión de modificar en esta fase o documentar con los diversos instrumentos de la programación extrema que los explicaremos más adelante, y que los podremos tener en cuenta para la fase de mantenimiento.

2.4.4 Fase de mantenimiento

La fase de mantenimiento requiere de un mayor esfuerzo de los desarrolladores para satisfacer algunas tareas del cliente. En esta fase se suele incorporar nuevo personal y tener un cambio en la estructura del equipo.

En esta entapa debemos tener claro los siguientes puntos:

- Mayor esfuerzo de trabajo
- Disminuye la velocidad del desarrollo
- El grupo de trabajo puede requerir de incorporación de más gente

2.4.5 Fase de muerte

Es cuando el cliente está satisfecho en cuanto a las necesidades y requerimientos planteados y ya no tiene más historias para incluirlas en el sistema.

En este punto se destacan los siguientes aspectos:

- El cliente no tiene más historias para incluir en el proyecto
- Se genera la documentación final del proyecto y ya no se dan más cambios en la arquitectura del mismo.

Vale destacar que también se da la muerte del proyecto cuando el sistema no cumple las expectativas del cliente o cuando no existen recursos para mantenerlo.

2.5 **Practicas de XP**

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 07 – 03 – 2010
Descripción: Prácticas de programación extrema.

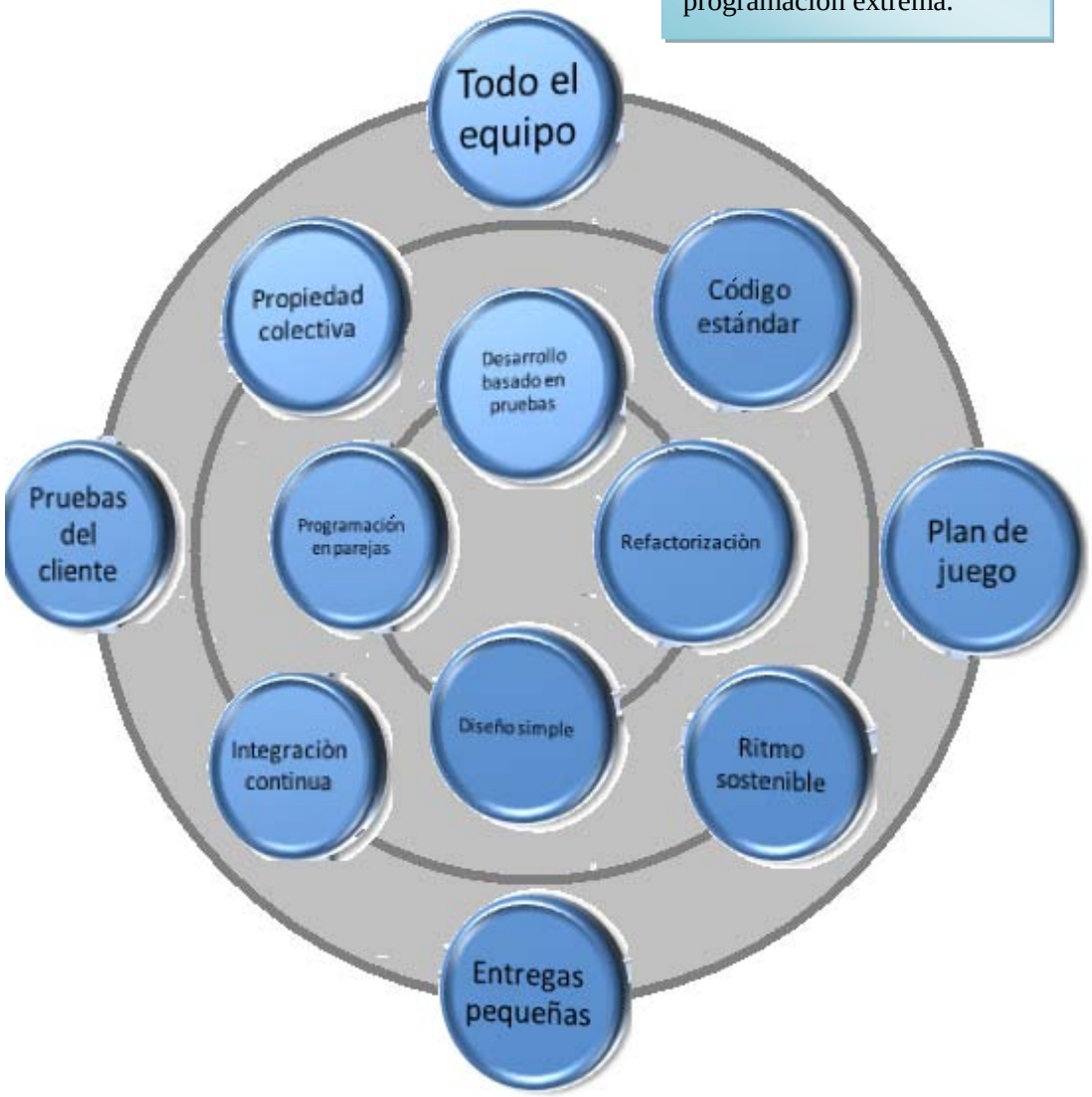


Figura #13
Prácticas de XP.

2.5.1 El juego de planificación:

El plan de juego consiste en estimar cuanto tiempo nos vamos a tardar en cada módulo, quedar de acuerdo en fechas de entrega de cada avance del proyecto y definir las técnicas a usarse.

2.5.2 Diseño simple:

La principal característica del diseño simple es no implementar características que no son necesarias, solo las indispensables. Con esta técnica las clases pueden ser descubiertas durante su análisis y ver cuáles son realmente necesarias para el sistema.

2.5.3 Entregas pequeñas:

Cuando se realiza un sistema se producen periódicamente nuevas versiones, corrigiendo errores o a la vez agregando al programa nuevas necesidades que tenga el cliente.

2.5.4 Pruebas continuas:

Las pruebas son un elemento esencial dentro de la programación extrema. Estas pruebas garantizan que las modificaciones en el código no rompan su funcionalidad e incluso puedan servir como estrategia de diseño escribiendo los casos de pruebas antes que el código.

2.5.5 Refactorización:

La refactorización consiste en modificar el código del sistema sin cambiar el funcionamiento del mismo. Esto se lo puede hacer simplificando las funciones, eliminando código que este duplicado o en último de los casos si el código está muy complicado podríamos modificar el diseño y hacerlo más fácil.

2.5.6 Programación en parejas:

Es trabajar conjuntamente para sacar al proyecto adelante. Esto ayuda para agilitar el proceso de programación ya que se trabaja en un solo equipo y se comparte los

conocimientos de cada uno para un objetivo en común y así se conseguirá un producto de calidad y con ahorro de tiempo.

2.5.7 Integración continua:

Antes y después de la integración del código debemos pasar por los casos de prueba, ya que las modificaciones se dan frecuentemente en el código base, esto quiere decir que para la integración se necesita realizar pruebas funcionales en las que participa el cliente.

2.5.8 Posesión colectiva del código:

El programador tiene la potestad de manipular y modificar cualquier parte del sistema siempre y cuando se sigan los estándares ya impuestos por los programadores y el cliente, además las pruebas se deben realizar cuando exista una modificación o integración en el código para evitar posibles errores.

2.5.9 Código estándar:

El código que se va a escribir debe estar definido y establecido a un estándar de codificación.

2.5.10 Todo el equipo:

El trabajo en equipo es fundamental para este tipo de programación, ya que se facilita el trabajo, se producen menos errores, va de acuerdo a las necesidades del consumidor final y se reduce el periodo de tiempo en el desarrollo del software.

2.5.11 Pruebas con el cliente:

El cliente debe estar comprometido a trabajar conjuntamente con los desarrolladores, responder preguntas, fijar prioridades, etc. La idea es tener un cliente que nos proporcione el mayor tiempo posible y aporte con los conocimientos de su área para así formar un buen equipo de trabajo y lograr un producto de calidad.

2.5.12 Ritmo sostenible:

El ritmo de trabajo promedio es de no más de 40 horas por semana, si se va hacer horas extras se recomienda que no sean a semana seguida por lo que esto afecta en el rendimiento del personal y por ende a la calidad del producto.

2.6 Instrumentos usados en la programación extrema

Los principales instrumentos que se usan en la programación extrema son:

- Las historias de usuario

  Son una pequeña descripción del comportamiento del sistema, no se necesita lenguaje técnico y usa terminología del cliente.

  Estas historias cuentan con tres aspectos fundamentales: la tarjeta la cual detalla la historia, la conversación entre cliente y desarrolladores, y las pruebas de aceptación, las mismas que sirven para ver si la historia ha sido correctamente implementada.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 11 – 03 – 2010
Descripción: Tabla historia de usuario
Fuente:http://www.scribd.com/doc/26495

HISTORIA DE USUARIO	
Numero:	**Nombre de historia de usuario:**
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:**
Prioridad en negocio: (Alta/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio/Bajo)	**Puntos reales:**
Descripción:	
Observaciones:	

Tabla #2
Historia de Usuarios

- Casos de prueba

  Cada historia tiene sus casos de prueba. Cada caso está compuesto de la prueba y el resultado. A este resultado se le hace una evaluación para determinar qué cambios se puede efectuar. Si estas pruebas son satisfechas se dice que la historia está completa.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 11 – 03 – 2010
Descripción: Tabla casos de prueba.
Fuente:http://www.scribd.com/doc/26495

CASOS DE PRUEBA	
Número caso de prueba:	**Número historia de usuario:**
Nombre caso de prueba:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Resultado esperado:	
Evaluación:	

Tabla #3
Tabla de Casos de Prueba.

- Las tareas de ingeniería

  Las historias se dividen en tareas, los pasos necesarios para poder implementar la historia. Los desarrolladores se encargan de realizar estas tareas, ya que se designa un programador responsable para cada tarea sea esta desarrollo, corrección o mejora.

Las tareas de ingeniería presentan el siguiente formato:

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 11 – 03 – 2010
Descripción: Tabla tarea de ingeniería.
Fuente:http://www.scribd.com/doc/26495149/Programacion-extrema-Informe

TAREA	
Numero tarea:	**Numero historia:**
Nombre de tarea:	
Tipo de tarea : Desarrollo/Corrección/Mejora/Otra	**Puntos Estimados:**
Fecha inicio:	**Fecha fin:**
Programador responsable:	
Descripción:	

Tabla #4
Tabla de tarea de ingeniería.

- Tarjetas CRC[24]

  Estas tarjetas se dividen en 3 partes, la primera que contiene el nombre de la clase, puede ser persona, cosa, evento, pantalla o concepto. La segunda son las responsabilidades de la clase y por último la tercera son los colaboradores, los cuales son las clases que trabajan en conjunto para ejecutar una responsabilidad.

[24] (clase, responsabilidad y colaboración). Son una metodología para el diseño de software

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 11 – 03 – 2010
Descripción: Tarjeta CRC.
Fuente:http://www.scribd.com/doc/2649

NOMBRE DE LA CLASE	
Responsabilidades	**Colaboradores**

Tabla #5
Tarjetas CRC.

2.7 Escenarios en programación extrema

En la programación extrema todos los requerimientos que son dados por el cliente se los toma como escenarios o también llamados historias de usuario en las cuales describe las necesidades que posteriormente serán modeladas e implementadas por los desarrolladores.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 07 – 03 – 2010
Descripción: Escenarios de programación

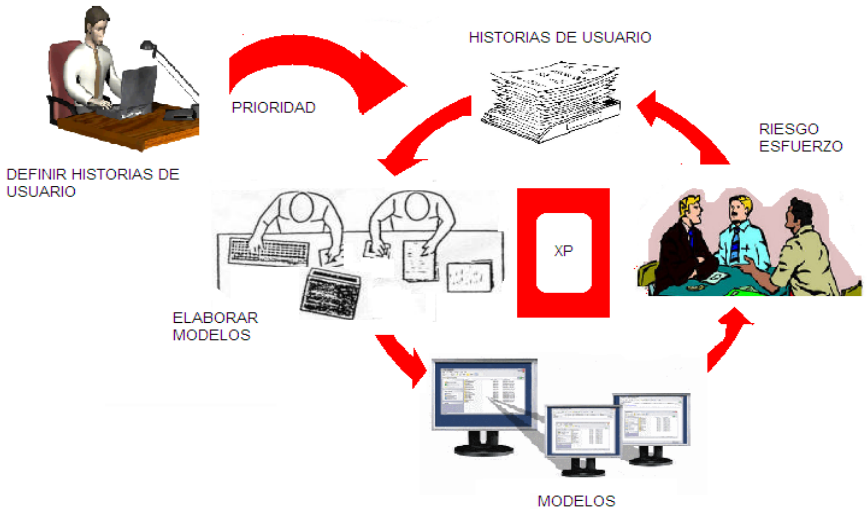


Figura #14
Escenario de XP

CAPÍTULO III: Levantamiento de requerimientos

3.1 Requerimientos para la web

En la actualidad ha surgido un gran avance en las comunicaciones y el internet, esto ha hecho que exista un gran interés en la elaboración de propuestas metodológicas, las que pueden ofrecer una óptima moldura de referencia al momento de desarrollar un sistema web.

3.1.1 La ingeniería de requisitos

Para el desarrollo de cualquier sistema es de suma importancia la identificación de requisitos. Definir las necesidades del sistema es un proceso complejo, ya que debemos identificar los requisitos del sistema para que cumpla las expectativas del cliente.

Este proceso de identificación de requisitos se lo divide en 3 actividades que son captura de requisitos, definición de requisitos y validación de requisitos.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 01 – 06 – 2010
Descripción: Proceso de ingeniería de requerimientos.
Fuente: http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf

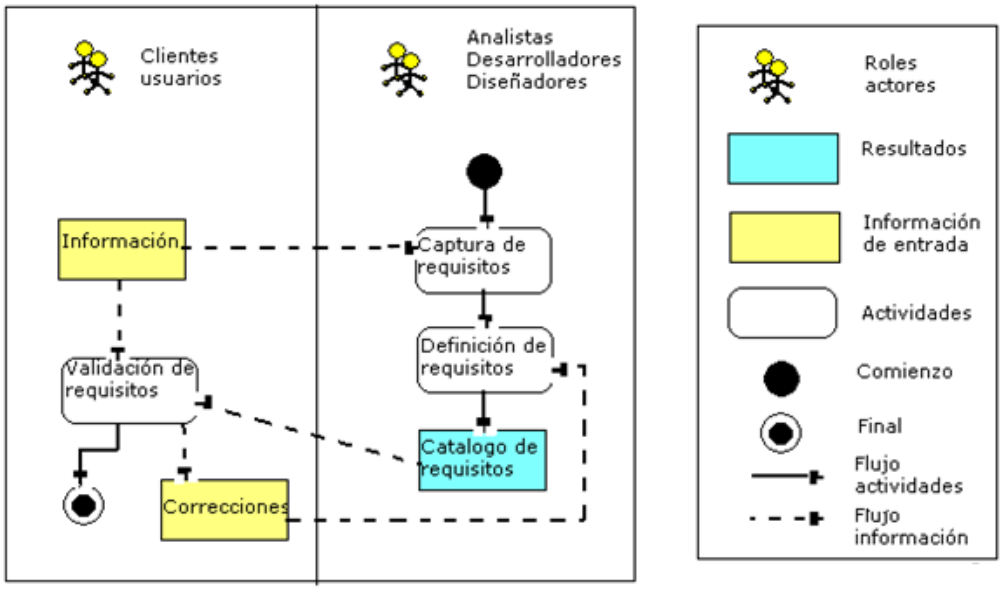


Figura #15
Proceso de ingeniería de requerimientos

En la figura anterior podemos ver el proceso de ingeniería de requerimientos, la que comienza con la captura de requisitos, luego el grupo de técnicos toma la información que

está dada por documentos, aplicaciones existentes, entrevistas, etc. que es proporcionada por los usuarios y clientes. En base a esta información los desarrolladores se encargan de elaborar el catálogo de requisitos, los cuales serán valorados, y de ser necesario se harán las debidas correcciones.

3.2 Metodología para el Desarrollo del Sitio Web

3.2.1 NDT (Navigational Development Techniques)

Es una metodología para el desarrollo web la cual se centra en la ingeniería de requisitos. El proceso de NDT consiste en definir los objetivos y en base a estos se pueden detallar los diferentes requisitos del sistema los cuales según esta metodología se los divide en:

3.2.1.1 Requisitos de almacenamiento de información

En este punto se definirá cual es la información que se va a manejar y la relación que existe entre los datos obtenidos.

3.2.1.2 Requisitos de actores

Aquí se definen los diversos roles que interactuaran en el sistema y las relaciones que pueden existir entre ellos.

3.2.1.3 Requisitos de interacción

Aquí podremos determinar cómo se va a mostrar la información y como se podrá ver en el sistema.

3.2.1.4 Requisitos funcionales

Este requisito es el más importante ya que se define la funcionalidad que tiene del sistema.

3.2.1.5 Requisitos no funcionales

Estos son los que se encargan de recolectar otros requisitos del sistema, el siguiente pasó luego de la validación de los requisitos es el de generar los siguientes 3 modelos

3.2.1.5.1 Modelo conceptual

Es la representación de la estructura del sistema mediante un diagrama de clases.

3.2.1.5.2 Modelo de navegación[29]

Aquí podemos ver como se podrá navegar en el sistema por medio de un conjunto de diagramas de clases.

3.2.1.5.3 Modelo de interfaz abstracta

En este punto se enseñará cómo va a interactuar el usuario con el sistema mediante la construcción de prototipos valorables.

3.2.2 SOHDM (Scenario-based Object-Oriented Hypermedia Design Methodology)

Es una de las primeras metodologías que apareció para la web. SOHDM propone que a partir de un modelo conceptual se lo representará con un diagrama de clase, el modelo se mantendrá agrupado y así se podrá conseguir que éstas formen un modelo de clases navegacionales del sistema.

Esta propuesta se divide en seis fases:

3.2.2.1 Fase de Análisis.

En ésta fase se estudiará las necesidades del sistema, se verá cómo está conformado el entorno de trabajo donde se realizará el sistema y conseguir los diferentes escenarios de las actividades que se pueden tomar en cuenta para el sistema.

3.2.2.2 Fase de Modelado de Objetos.

Se elabora el diagrama de clases el cual nos permitirá ver la estructura conceptual del sistema

3.2.2.3 Fase de Diseño de Vistas.

Para obtener una vista de los objetos, se trata de agrupar los objetos en unidades navegacionales.

Se reorganizan los objetos en unidades navegacionales que representan una vista de los objetos del sistema

[29] Un salto de una página web a otra página web, provocado por la selección de un enlace, en el que se cambia de contenido (información y/o funcionalidad

3.2.2.4 Fase de Diseño Navegacional.

Se definen enlaces e hiperenlaces para crear un ambiente amigable.

3.2.2.5 Fase de Diseño de la Implementación.

Se diseña la base de datos, las páginas, la interfaz del sistema

3.2.2.6 Fase de Construcción.

Luego de los diseños se procede a la construcción del sistema, elaborando la base de datos y posteriormente la implementación.

3.2.3 OOWS (Object Oriented Web Solution – Soluciones Web Orientadas a Objetos

OOWS, es un método para la construcción de aplicaciones Web, este proceso utiliza la notación UML para el desarrollo de un sitio web.

Para este proceso se generan 3 modelos que son los siguientes:

3.2.3.1 Modelo estructural.

Para este modelo se utiliza los diagramas UML, con los cuales se puede definir las clases y las relaciones que existen entre sí.

3.2.3.2 Modelo dinámico.

Para este modelo se utiliza los diagramas de transición de estados y los diagramas de secuencia UML para describir la interacción que existe entre los objetos de diferentes clases.

3.2.3.3 Modelo funcional.

En este modelo se especifica de forma textual si existe un cambio de estado del objeto, notificando el efecto de sus servicios.

3.2.4 OOHDM (Object Oriented Hypermedia Design Model)

Es un proceso para la elaboración de aplicaciones web el cual se basa en prototipos de desarrollo interactivo y de desarrollo incremental. OOHDM propone el desarrollo de aplicaciones web a través de un proceso compuesto por 4 etapas:

3.2.4.1 Diseño conceptual.

El esquema de las clases consiste en el diseño de un conjunto de clases con las relaciones que existe entre ellas, para esto podemos usar lenguaje UML, tarjetas de clase y tarjetas CRC.

3.2.4.2 Diseño navegacional.

El diseño navegacional consiste en construir una vista sobre un diseño conceptual proporcionando un modelo más amigable.

3.2.4.3 Diseño de interfaces abstractas.

En esta etapa se especificará la organización y el comportamiento de la interfaz del sistema.

3.2.4.4 Implementación.

En esta fase se toma en cuenta el entorno en donde se va a correr el sistema, Se analiza la información que debe ser almacenada y al igual que el diseño de interfaces abstractas se ve el comportamiento de la interfaz, a fin de que la aplicación se encuentre bien organizada.

La metodología que vamos a usar es NDT ya que utilizamos programación extrema y éste método es el más adecuado para esta técnica de programación, además NDT es un entorno metodológico compuesto por un conjunto de técnicas para capturar los requisitos y modelar los aspectos propios de los sistemas web de una manera sencilla, pero a la vez de gran calidad.

3.3 PHP

3.3.1 ¿Qué es el PHP?

PHP es un lenguaje de script[30] que se lo inserta dentro del HTML y ejecutado en el lado del servidor. La meta de este lenguaje es desarrollar páginas dinámicas, las cuales tengan comunicación con servidores de aplicaciones para obtener información de base de datos, que permitan hacer validaciones, almacenar información y realizar actualizaciones.

[30] Código fuente de un programa escrito en algún lenguaje interpretado.

La más importante de las características de PHP es el soporte para un sinnúmero de bases de datos, entre ellas podemos mencionar InterBase, MySQL, Oracle, PosgreSql, entre otras. Además en PHP podemos integrar varias bibliotecas externas las cuales ayudan al desarrollador a generar diversas aplicaciones.

Una de las ventajas de usar PHP es que es un producto de código abierto, lo cual es de gran ayuda, ya que existe un gran grupo de programadores que están pendientes del código y si encuentran fallas en el funcionamiento son rápidamente reparadas, teniendo así una mejora continua del producto.

3.3.2 Qué podemos hacer con PHP

PHP es uno de los lenguajes de programación web más utilizados en la actualidad, con el cual podemos realizar programas, aplicaciones web, servicios del lado del servidor, etc. Y entre sus múltiples funciones, PHP puede generar imágenes, generar vistas miniaturas, generar archivos pdf, y muchas otras opciones que con un buen conocimiento del lenguaje se las puede explotar.

3.3.3 Requisitos para iniciar a programar con PHP

Para poder iniciar a programar con PHP, necesitamos instalar ciertas herramientas en nuestra PC que nos van a ayuda en el desarrollo del sistema como un editor de texto y un servidor, ya que es un lenguaje que funciona del lado del servidor.

PHP se lo diseñó para ejecutarse junto a Apache, que es un servidor web y se lo utiliza en sistemas operativos Linux pero actualmente se los utiliza con Mac OS y Windows como es nuestro caso.

3.3.4 Inicio de un script en PHP

El ´<?PHP´, al inicio indica lo que va a ser interpretado por PHP y al final se pone un ´?>´ para indicar la terminación del mismo. Esto quiere decir que lo que este fuera de este límite será parte del código HTML.

3.3.5 Variables

Las variables son la base de cualquier lenguaje de programación. Las Variables en PHP están representadas por un signo de dólar "$" y seguido por el nombre de la variable.

Dependiendo de la información que contenga una variable, esta puede ser considerada de uno u otro tipo:

Variables numéricas:

Enteros → $entero=2010;

Real→ $real=3.14159;

Variables alfanuméricas:

Cadenas→$cadena="Figuras";

Tablas:

```
$figura[1]="cuadrado";
$figura [2]="circulo";
$figura [3]="rombo";
$figura [4]="ovalo ";
$figura [5]="trapecio";
```

3.3.6 Función Print

La función "print" es fundamental para la salida de datos ya que manda cualquier cosa, sea esta entre comillas o paréntesis. Lo mismo sucede con la función "echo", pero la diferencia es que print permite ver al desarrollador si la salida se a realizado con éxito o no.

```
print 'Hola, mundo';
print('Hola, mundo');
```

3.4 Calidad de producto

La calidad de un producto de software se la obtiene luego de pasar por una serie de revisiones y pruebas que se hace a lo largo del desarrollo, con esto se asegura que se incrementará la fiabilidad, se reducirá el mantenimiento, se mejorará la dirección del proyecto y se podrá detectar errores pronto.

Además para obtener un producto de calidad es indispensable conocer al cliente, sus necesidades y su entorno, con la finalidad de optimizar la producción del software y aumentar la satisfacción del cliente.

3.4.1 ISO/IEC 9126

Tecnologías de la Información – Calidad de los productos software.

Es un modelo normalizado para evaluar el software el cual está dividido en 4 partes:

3.4.1.1 Modelo de calidad

Clasifica la calidad del software en un conjunto estructurado de características y sub-características que se basa en 6 aspectos:

Funcionalidad

Fiabilidad

Usabilidad

Portabilidad

Mantenibilidad

Eficiencia.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 26 – 06 – 2010
Descripción: Modelo de calidad.
Fuente: http://www.myt.com.pe/principal/images/9126_3.png

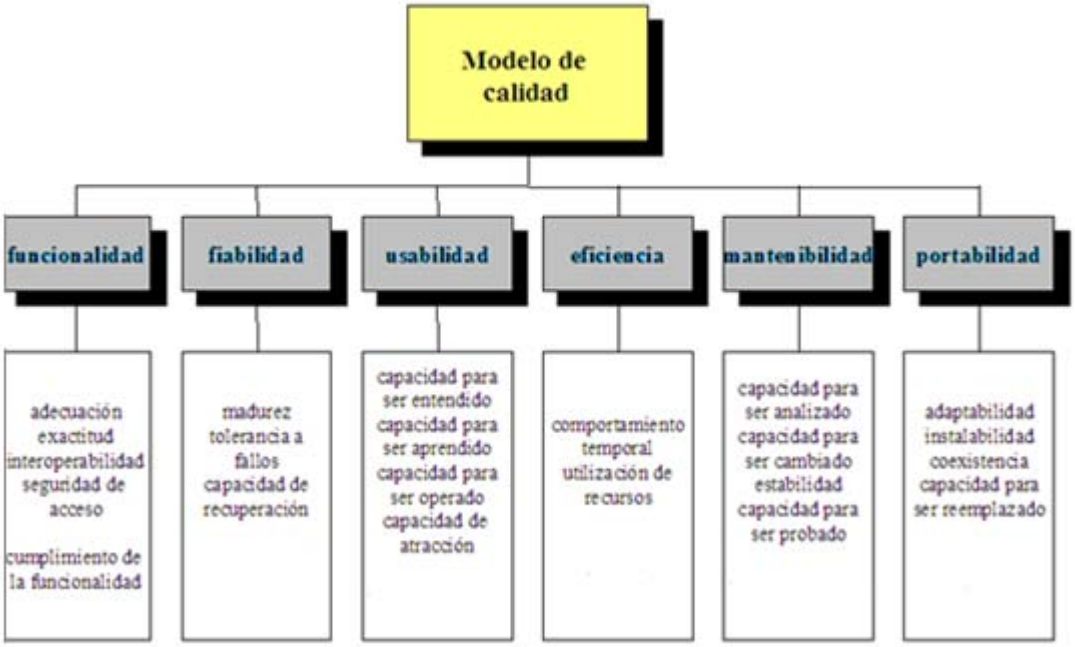


Figura #16
Modelo de Calidad

Como podemos ver en la figura el ISO 9126 nos permite definir un modelo de calidad, en base a las 6 características expuestas. El modelo de calidad que definamos nos dará como resultado el grado de calidad de cada uno de nuestros productos software.

3.4.1.1.1 Funcionalidad

Conjunto de características y atributos relacionados con las funciones propiedades especificas del sistema. Existe coherencia entre las necesidades detectadas y los resultados que se obtiene.

3.4.1.1.1.1 Adecuación

Se refiere al grado de adaptación del usuario con el software, proporcionando un conjunto apropiado de funciones para realizar tareas.

3.4.1.1.1.2 Exactitud

Se obtienen los resultados o efectos correctos o acordados de una forma precisa.

3.4.1.1.1.3 Interoperabilidad

La interoperabilidad es la condición mediante la cual se puede interactuar con uno o más sistemas especificados.

3.4.1.1.1.4 Seguridad

Se refiere al acceso seguro a un sistema, protegiendo la información de personas ajenas o sistemas no autorizados.

3.4.1.1.1.5 Cumplimiento de normas

El producto de software debe regirse a ciertas normas y regulaciones de ley que se asemejen a la funcionalidad del producto de software.

3.4.1.1.2 Fiabilidad

Un conjunto de características relacionadas con la capacidad del software de mantener el nivel requerido en condiciones determinadas durante un cierto intervalo de tiempo.

3.4.1.1.2.1 Madurez

Es la capacidad del sistema para mantenerse en funcionamiento a pesar de tener errores en el mismo.

3.4.1.1.2.2 Recuperabilidad

Es la recuperación de la información en caso de que el producto de software falle de alguna manera.

3.4.1.1.2.3 Tolerancia a fallos

Se refiere a que si existe un fallo en el sistema, se debe mantener en un nivel de funcionamiento específico.

3.4.1.1.3 Usabilidad

La usabilidad, hace referencia, a la rapidez y facilidad con que las personas realizan tareas especificas a través del uso de alguna herramienta o producto.

3.4.1.1.3.1 Aprendizaje

Se refiere a que el usuario puede aprender acerca del producto de software.

3.4.1.1.3.2 Comprensión

El producto de software da la posibilidad que el usuario entienda si el software es adecuado y como puede ser usado para diferentes tareas.

3.4.1.1.3.3 Operatividad

Capacidad del usuario para controlar y operar el producto de software.

3.4.1.1.3.4 Capacidad de atracción

Se refiere a que el producto de software debe ser atractivo al usuario.

3.4.1.1.4 Eficiencia

Conjunto de atributos relacionados con la precisión, plenitud y desempeño del software.

3.4.1.1.4.1 Comportamiento en el tiempo

Se refiere a que el producto de software debe poseer tiempo de respuesta y tiempo de proceso apropiados bajo ciertas condiciones.

3.4.1.1.4.2 Comportamiento de recursos

El producto de software debe usar los recursos necesarios para el funcionamiento óptimo bajo condiciones determinadas para sus funciones específicas.

3.4.1.1.5 Mantenibilidad

Facilidad con la que el software puede ser modificado para mejorar su funcionamiento y corregir posibles errores.

3.4.1.1.5.1 Estabilidad

Se refiere a que el producto de software debe ser capaz de evitar problemas si existe un cambio inesperado en el software.

3.4.1.1.5.2 Capacidad de análisis

El producto de software tiene la capacidad de identificar las deficiencias o posibles fallos del mismo.

3.4.1.1.5.3 Capacidad de cambio

Se refiere a que el producto de software puede aceptar cualquier modificación que se requiera.

3.4.1.1.5.4 Capacidad de pruebas

Se refiere a que cualquier actualización que exista en el producto de software, esta podrá ser validada.

3.4.1.1.6 Portabilidad

Conjunto de atributos relacionando la capacidad que posee un software para ejecutarse en diferentes plataformas.

3.4.1.1.6.1 Capacidad de instalación

El producto de software tiene la capacidad de ser instalado en un entorno específico.

3.4.1.1.6.2 Capacidad de reemplazamiento

Se refiere a que el producto de software se lo puede utilizar en lugar de otro producto, para un mismo propósito y entorno.

3.4.1.1.6.3 Adaptabilidad

El producto de software puede ser adaptado a diferentes entornos, sin la necesidad de aplicar acciones distintas.

3.4.1.1.6.4 Co-Existencia

Se refiere a que el producto de software puede co-existir con otro software similar, en un entorno común.

3.4.1.2 Métricas internas

Se aplican cuando el producto de software no está en ejecución y se lo mide por las propiedades estáticas del código, utilizando técnicas de inspección.

3.4.1.3 Métricas externas

Las métricas externas son las que se aplican cuando el software está en ejecución y se mide por las propiedades dinámicas del código.

3.4.1.4 La calidad en las métricas de uso

Las métricas de uso se las utiliza cuando el producto de software está siendo usado en condiciones reales. Se mide por el grado en el cual el software está realizado en función de las necesidades del usuario en el entorno de trabajo para el que fue construido.

En el siguiente grafico podemos apreciar lo mencionado anteriormente:

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 27 – 09 – 2010
Descripción: Aspectos de calidad iso 9126
Fuente: http://www.myt.com.pe/principal/Imp_9126.php

Figura #17
Aspectos de calidad ISO 9126

3.5 ESTÁNDAR PARA LA WEB

www.W3C.org

La World Wide Web Consortium (W3C), es la principal organización de normas internacionales para la (WWW o W3 abreviada).

3.5.1 ¿Qué son los Estándares Web?

Entendemos por estándar el conjunto de normas y reglas, las cuales describen los requisitos a cumplirse en el producto de software, esto es necesario para que los diferentes elementos de hardware y software que se vayan a utilizar, sean compatibles entre sí.

Existe una organización que se encarga de desarrollar estándares relacionados con la web (W3C). Estos estándares nos sirven para tener una referencia para la construcción de una web que sea accesible, interoperable[31] y eficiente, con lo cual podremos desarrollar cada vez aplicaciones de mayor calidad.

La W3C se encuentra conformada por un gran grupo de colaboradores en todo el mundo que dan recomendaciones para lograr que esta organización obtenga los resultados esperados. Entre ellos tenemos grandes empresas de hardware y software, universidades, administraciones públicas y diferentes usuarios que quieran mostrar su opinión.

Algunos de los estándares Web más conocidos y ampliamente utilizados son: HTML (HyperText Markup Language), para definir la estructura de los documentos; XML (eXtensible Markup Language), que sirve de base para un gran número de tecnologías; y CSS (Cascading Style Sheets), que permite asignar estilos para la representación de los documentos.

El objetivo de utilizar estándares para la web, es conformar una web universal, la cual sea fácil de usar, sea accesible y confiable para todo el mundo. Con esto se pretende mejorar la infraestructura para poder obtener una web que tenga organizada su información y sea más entendible para el usuario.

[31] Condición mediante la cual sistemas heterogéneos pueden intercambiar procesos o datos.

3.5.2 Acceso Universal

Es la opción que tiene el usuario para acceder a la web desde cualquier lugar, cualquier momento y desde cualquier tipo de dispositivo, para esto se toma en cuenta múltiples factores como el idioma, la localización geográfica, posibles limitaciones físicas, psíquicas o sensoriales del usuario.

3.5.3 Confianza en la Web

La web es un sitio donde diariamente interactúan millones de usuarios, loscuales realizan transacciones, crean contenidos, crean redes sociales, etc.

Para esto existen algunas tecnologías, las cuales se las emplea para que los usuarios puedan confiar en el sitio web, entre las cuales podemos destacar las firmas digitales de documentos, encriptación de los datos para la confidencialidad; y mecanismos de establecimiento y declaración de las políticas de privacidad de los datos de los sitios Web.

3.5.4 ¿Cómo funcionan?

El proceso de la creación de un estándar Web es elaborado por especialistas en la materia con la ayuda de todos los usuarios que quieren aportar sus opiniones y conocimientos para lograr obtener estándares de calidad. Estos estándares están sujetos a la Política de Patentes W3C, lo que permite que sean utilizados libremente por todo el mundo.

La creación de estos estándares favorecen a todos los usuarios, ya que al utilizar las mismas tecnologías, el usuario puede interactuar con el resto y tener la capacidad de que las maquinas se comprendan entre sí, como lo podemos apreciar en el siguiente grafico.

Modificado por: Landázuri Raúl, Siza Richard.
Fecha: 27 – 07 – 2010
Descripción: Distintos dispositivos y redes comunicándose a través de protocolos y lenguajes comunes
Fuente:http://www.w3c.es/divulgacion/guiasbreves/images/estandar

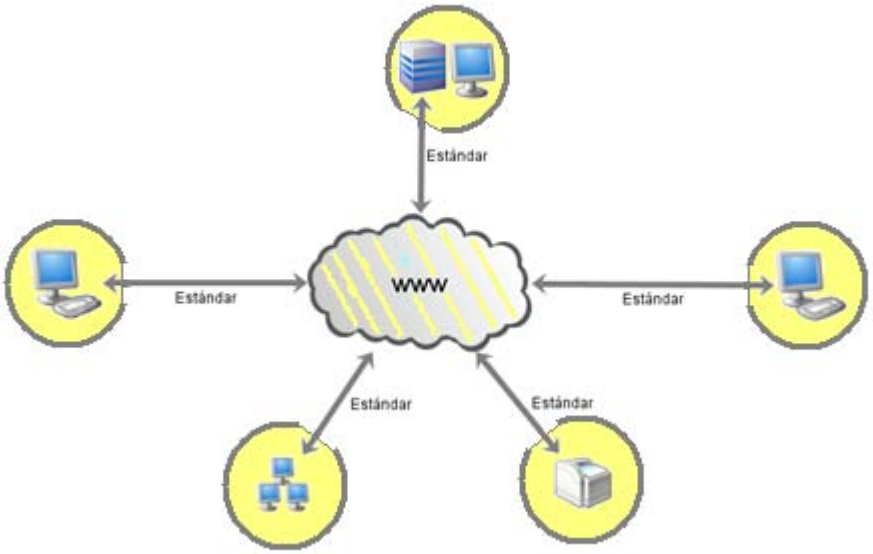


Figura #18
Comunicación de dispositivos por la red

3.5.5 Herramientas

El W3C ofrece una serie de herramientas que permiten verificar si se hace una correcta aplicación de las especificaciones, como por ejemplo manuales de directivas, buenas prácticas de tecnologías concretas y los validadores sintácticos de los lenguajes.

Entre las principales herramientas tenemos las hojas de estilo en cascada (CSS) las cuales definen cómo se van a presentar los elementos, es decir, qué colores, fuentes de letra, márgenes, alineaciones, etc.

3.5.6 ¿Qué es CSS?

Hojas de Estilo en Cascada es una forma o mecanismo de presentar un documento en la pantalla. Esto ayuda a los desarrolladores a tener un control sobre el diseño y estilo para la presentación del sitio.

3.5.6.1 Usar CSS para diseñar la presentación del contenido

Antiguamente el diseño de las páginas se basaba en la edición infinita de líneas en HTML que debía ser reutilizado página a página en el web site.

Si cambiamos el antiguo método y accedemos a las CSS se ganara 2 cosas:

3.5.6.1.1 Tener mayor agilidad en el proceso de crear un nuevo contenido y en la actualización y mantenimiento de un contenido creado.

La agilidad de la producción es algo fundamental y lo podemos lograr a través de los CCS, ya que se podrá evitar los cuellos de botella dividiendo la creación del contenido de la presentación.

Se logrará un mejor funcionamiento de los servidores, debido a que los documentos se podrán descargar con mayor velocidad.

Se logra una mayor velocidad de bajada con la eliminación total de código de presentación dentro del HTML del contenido y evitando generar tablas en cada página, esto se lo puede conseguir con las CSS, las cuales evitan toda esta redundancia.

3.5.6.2 Validar con w3.org

Validar según las normas de w3.org no es fácil. La mejor táctica es usar CSS que validen y de ahí modificar lo que sea necesario para acomodar el estilo a nuestras preferencias.

3.5.7 Lenguaje CSS

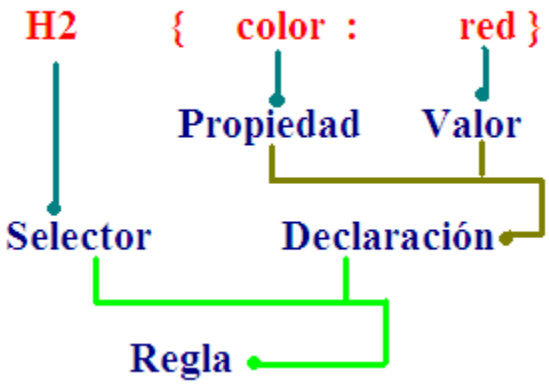
Una hoja de estilo está compuesta por algunas reglas, tales como:

3.5.7.1 Selector

El cual indica a qué elementos se refiere la regla

3.5.7.2 Declaraciones

Cada declaración contiene propiedad:valor los que indican cómo se despliegan los elementos a los que se refiere el selector.



Podemos asociar una misma declaración a diferentes selectores usando comas entre ellos, usando varias declaraciones a un selector utilizando punto y coma o realiza las dos agrupaciones a la vez como mostramos en los ejemplos respectivamente.

H1, H2, H3, H4, H5, H6 { color : red }

H1 { color: red; text-align: center; }

H1, H2, H3, H4, H5, H6 { color : red; text-align: center; }

Los comentarios dentro de las hojas de estilo comienzan con "/*" ,"(contenido)" y terminan con "*/".

H2 { color: blue } /* Los subtítulos van en azul */ .

3.5.7.3 Tipos de letra

Es un verdadero problema la especificación del tipo de letra en el World Wide Web, ya que sus nombres varían de una plataforma a otra; y no es poco común encontrar que bajo una misma plataforma existan tipos que estén en una versión y no en otra.

La solución podría ser que cada vez que se especifique algún tipo de letra, se tenga varias alternativas, ya que hay casos que en el computador del usuario no exista el tipo de letra especificado, ahí se prueba con el siguiente, y así sucesivamente hasta llegar al último. En caso de que el último tipo de letra

tampoco sea reconocido, se utiliza el valor por omisión para el browser, que comúnmente es Times New Roman.

Por ejemplo:

P { font-family: "Arial", "Helvetica", sans-serif }

Esto quiere decir que a todos los párrafos se les asignará el tipo de letra Arial (este es un tipo de letra de MS Windows), en caso de no estar disponible en el sistema se utilizará Helvetica (existente bajo MacOS).

El último valor que esta puesto en el ejemplo merece una explicación.

sans-serif (sin adornos) serif (con adornos) cursive (manuscrito monospace (monoespaciado, o espacio constante).

Esto es necesario para evitar que el browser caiga en el tipo de letra por omisión.

3.5.7.4 Colores

Los colores se indican mediante un código RGB, (ejemplo: #FFEE09), o bien mediante uno de los nombres de color estandarizados:

aqua: Agua

black: Negro

blue: Azul

fuchsia: Fucsia

gray: Gris

green: Verde

lime: Lima

maroon: Marrón

navy: Azul marino

olive: Verde oliva

purple: Púrpura

red: Rojo

silver: Plateado

teal: Verde azulado

white: Blanco

yellow: Amarillo

Los códigos RGB abreviados permiten escribir sólo 256 colores, esto es muy usual en hojas de estilo puesto que es difícil reconocer muchos matices en el texto, por lo que no tiene sentido ser tan específico en el color.

3.5.7.5 Longitudes

Las longitudes se indican con un número, seguido de un sufijo. Se pueden expresar en términos absolutos:

24mm: 24 milímetros

2cm: 2 centímetros

1.2in: 1,2 pulgadas

12pt: 12 puntos de impresión

3pc: 3 picas

105px: 105 pixeles

O bien en términos relativos al tamaño del tipo de letra actual:

2em: 2 veces el ancho de una letra "m"

3ex: 3 veces el ancho de una letra "x"

3.5.7.6 URL’s

Los URL se indican usando la palabra reservada "url", seguida de un paréntesis, luego el contenido del URL y luego un paréntesis de cierre. Como vemos en estos ejemplos:

.fondo1 { background: url(http://www.myhost.com/); }

```
.fondo2 { background: url(imagen.gif); }

.fondo3 { background: url(icn/globo.gif); }
```

3.5.7.7 Márgenes y bordes

Creado por: Landázuri Raúl, Siza Richard.
Fecha: 01 – 08 – 2010
Descripción: Márgenes y Bordes.

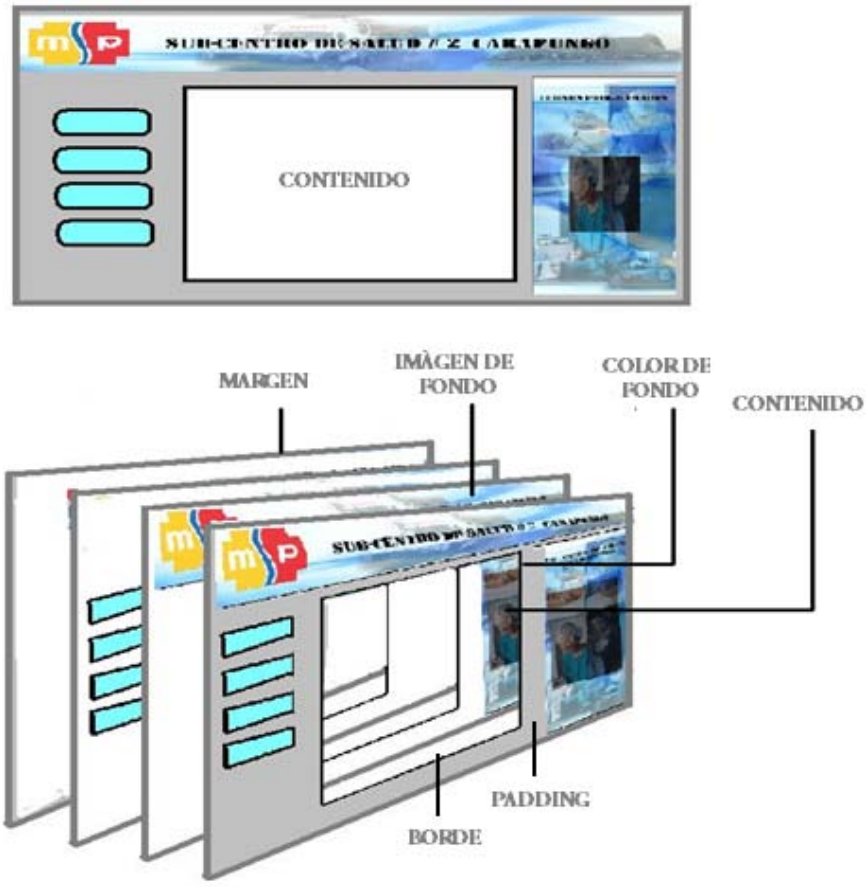


Figura #19
Márgenes y Bordes

Para tener una idea más clara de la nomenclatura que usamos en las hojas de estilo, explicaremos el significado y el comportamiento de algunos términos expuestos en la imagen anterior.

El padding es un espacio que forma parte del elemento, y separa la parte imprimible del selector (lo que llamamos, su contenido), de su borde.

El borde corresponde al borde visible que separa al espacio de padding del márgen.

El márgen es espacio que está fuera del elemento.

CAPÍTULO IV: Diagramación del sistema

4.1 Diagramas de Casos de Uso

Se los utiliza para obtener requisitos que el usuario necesita de forma general, es decir, se indica cómo debe el sistema interactuar con el usuario.

Los diagramas de casos de uso son muy útiles para describir sistemas interactivos (como es nuestro caso), porque nos enseña lo que el usuario pretende obtener del sistema cuando él lo use.

Creado por: Landázuri Raúl, Siza Richard.
Fecha: 16 – 10 – 2010
Descripción: Diagrama de Casos de Uso

Figura 20
Diagrama de Casos de Uso

4.1.1 Actor

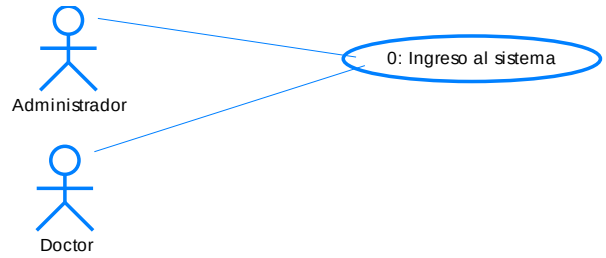
Actor es un rol que el usuario representa frente al sistema. El actor representa la labor o actividad que el usuario realiza en el sistema.

Para nuestro caso tenemos 2 actores claramente definidos:

- Administrador
- Doctor

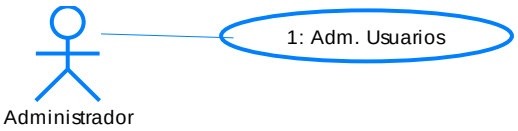
4.1.2 Descripción del diagrama de casos de uso

Escenario 0



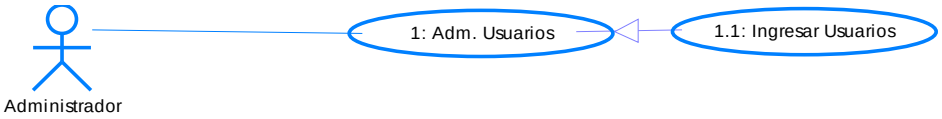
Caso de Uso:	Ingreso al sistema.
Actor:	Administrador, Doctor.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador o el Doctor pueden ingresar al sistema mediante el ingreso de un nombre de usuario y contraseña.

Escenario 1



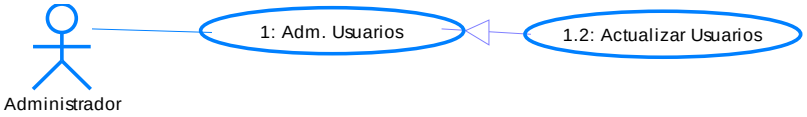
Caso de Uso:	Adm. Usuarios.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede administrar todos los Usuarios, es decir puede ingresar, actualizar, borrar y/o consultar.

Escenario 1.1



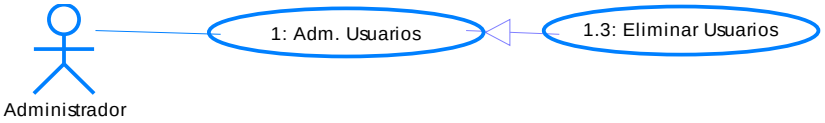
Caso de Uso:	Ingresar Usuarios.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede ingresar Usuarios nuevos al sistema con sus respectivos datos.

Escenario 1.2



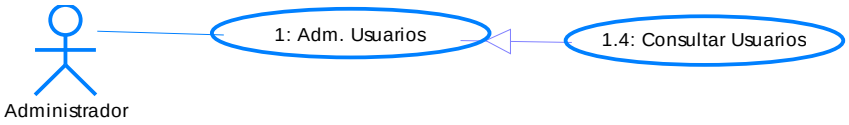
Caso de Uso:	Actualizar Usuarios.
Actor:	Administrador.
Precondiciones:	Tener Usuarios ingresados.
Tipo:	Primario.
Detalle:	El Administrador puede actualizar los datos de los Usuarios.

Escenario 1.3



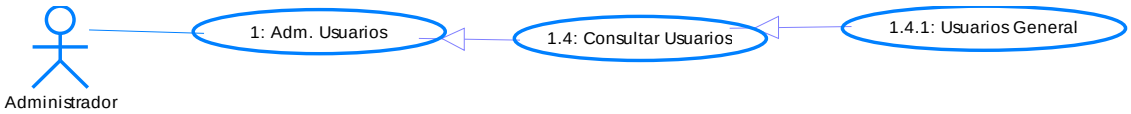
Caso de Uso:	Eliminar Usuarios.
Actor:	Administrador.
Precondiciones:	Tener Usuarios ingresados.
Tipo:	Primario.
Detalle:	El Administrador puede eliminar Usuarios.

Escenario 1.4



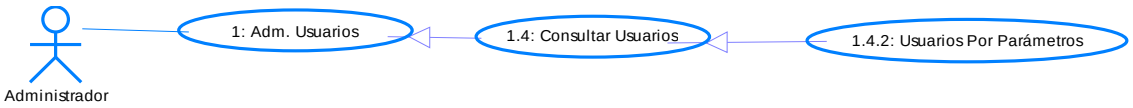
Caso de Uso:	Consultar Usuarios.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede consultar Usuarios mediante parámetros específicos o una búsqueda general.

Escenario 1.4.1



Caso de Uso:	Usuarios General.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador puede consultar Usuarios sin ningún parámetro específico.

Escenario 1.4.2



Caso de Uso:	Usuarios Por Parámetros.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador puede consultar Usuarios mediante parámetro específicos.

Escenario 2

Caso de Uso:	Adm. Doctores.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede administrar todos los Doctores, es decir puede ingresar, actualizar, borrar y/o consultar.

Escenario 2.1

Caso de Uso:	Ingresar Doctores.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede ingresar Doctores nuevos al sistema con sus respectivos datos.

Escenario 2.2

Caso de Uso:	Actualizar Doctores.
Actor:	Administrador.
Precondiciones:	Tener Doctores ingresados.
Tipo:	Primario.
Detalle:	El Administrador puede actualizar los datos de los Doctores.

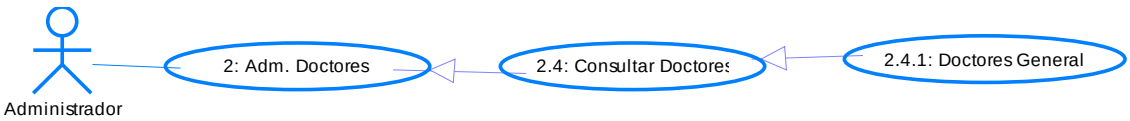
Escenario 2.3

Caso de Uso:	Eliminar Doctores.
Actor:	Administrador.
Precondiciones:	Tener Doctores ingresados.
Tipo:	Primario.
Detalle:	El Administrador puede eliminar Doctores.

Escenario 2.4

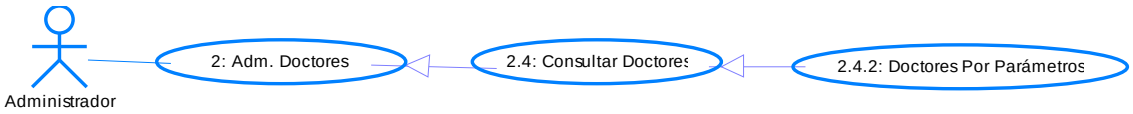
Caso de Uso:	Consultar Doctores.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede consultar Doctores mediante parámetros específicos o una búsqueda general.

Escenario 2.4.1



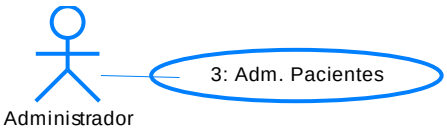
Caso de Uso:	Doctores General.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador puede consultar Doctores sin ningún parámetro específico.

Escenario 2.4.2



Caso de Uso:	Doctores Por Parámetros.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador puede consultar Doctores mediante parámetros específicos.

Escenario 3



Caso de Uso:	Adm. Pacientes.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede administrar todos los Pacientes, es decir puede ingresar, actualizar, borrar y/o consultar.

Escenario 3.1

Caso de Uso:	Ingresar Pacientes.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede ingresar Pacientes nuevos al sistema con sus respectivos datos.

Escenario 3.2

Caso de Uso:	Actualizar Pacientes.
Actor:	Administrador.
Precondiciones:	Tener Pacientes ingresados.
Tipo:	Primario.
Detalle:	El Administrador puede actualizar los datos de los Pacientes.

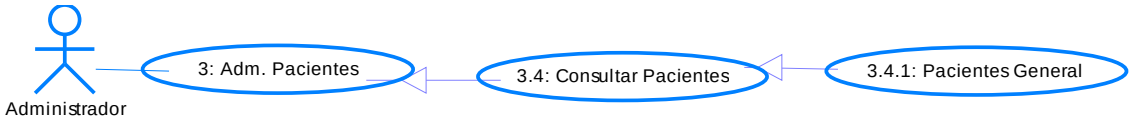
Escenario 3.3

Caso de Uso:	Eliminar Pacientes.
Actor:	Administrador.
Precondiciones:	Tener Pacientes ingresados.
Tipo:	Primario.
Detalle:	El Administrador puede eliminar Pacientes.

Escenario 3.4

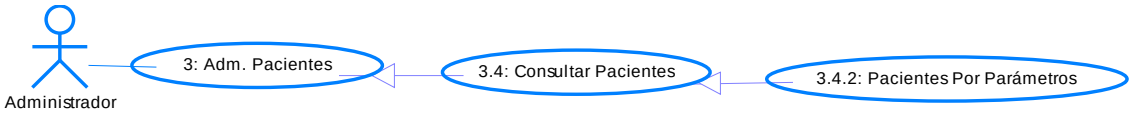
Caso de Uso:	Consultar Pacientes.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador puede consultar Pacientes mediante parámetros específicos o una búsqueda general.

Escenario 3.4.1



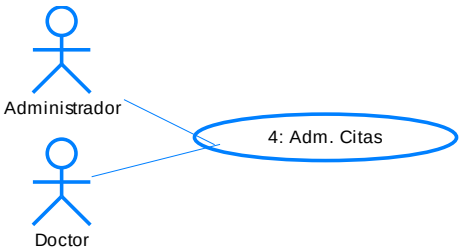
Caso de Uso:	Pacientes General.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador puede consultar Pacientes sin ningún parámetro específico.

Escenario 3.4.2



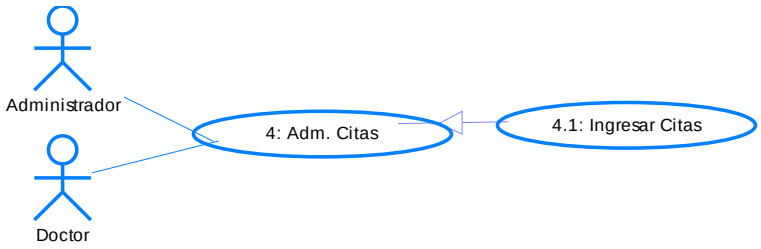
Caso de Uso:	Pacientes Por Parámetros.
Actor:	Administrador.
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador puede consultar Pacientes mediante parámetros específicos.

Escenario 4



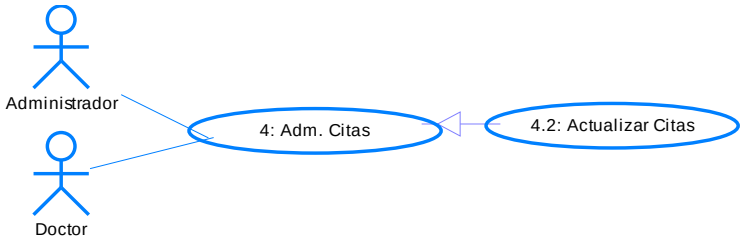
Caso de Uso:	Adm. Citas.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden administrar todas las Citas, es decir pueden ingresar, actualizar, borrar y/o consultar.

Escenario 4.1



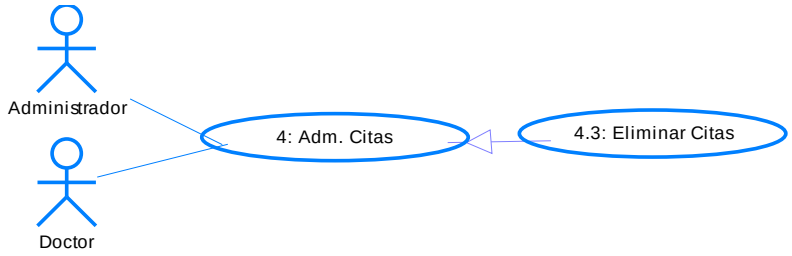
Caso de Uso:	Ingresar Citas.
Actor:	Administrador , Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden ingresar Citas nuevos al sistema con sus respectivos datos.

Escenario 4.2



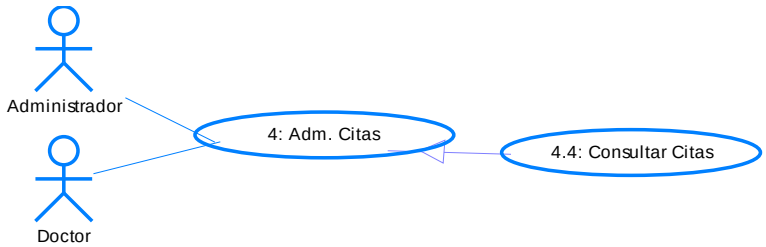
Caso de Uso:	Actualizar Citas.
Actor:	Administrador, Doctor
Precondiciones:	Tener Citas ingresadas.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden actualizar los datos de las Citas.

Escenario 4.3



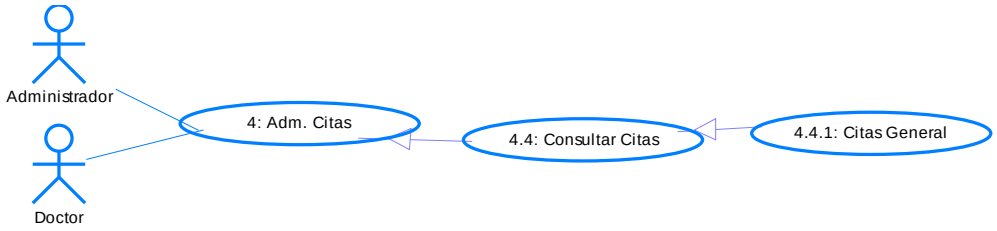
Caso de Uso:	Eliminar Citas.
Actor:	Administrador, Doctor
Precondiciones:	Tener Citas ingresadas.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden eliminar Citas.

Escenario 4.4



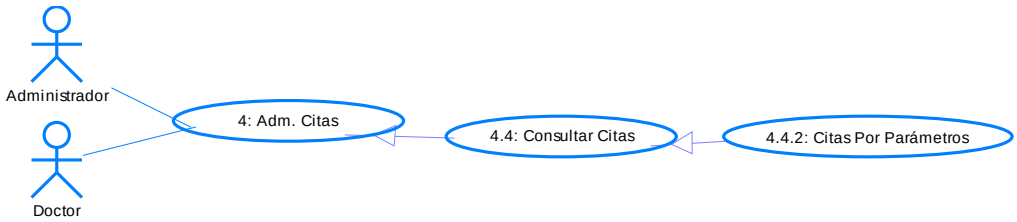
Caso de Uso:	Consultar Citas.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden consultar Citas mediante parámetros específicos o una búsqueda general.

Escenario 4.4.1



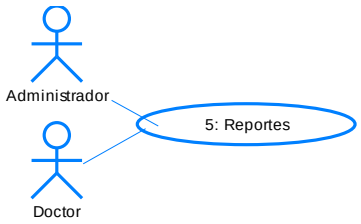
Caso de Uso:	Citas General.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador y/o Doctor pueden consultar Citas sin ningún parámetro específico.

Escenario 4.4.2



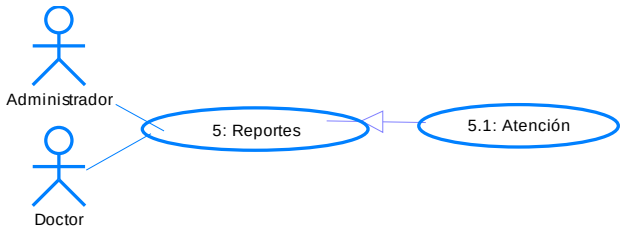
Caso de Uso:	Citas Por Parámetros.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador y/o Doctor pueden consultar Citas mediante parámetros específicos.

Escenario 5



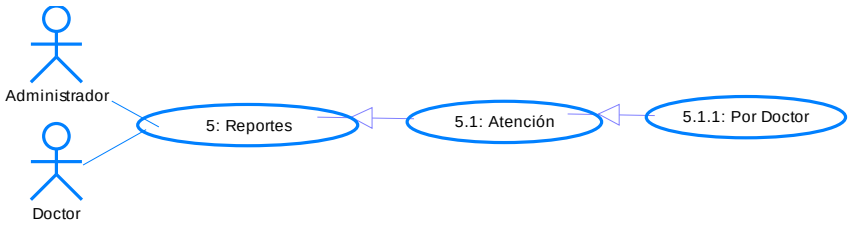
Caso de Uso:	Reportes
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden generar varios reportes según las opciones que escoja.

Escenario 5.1



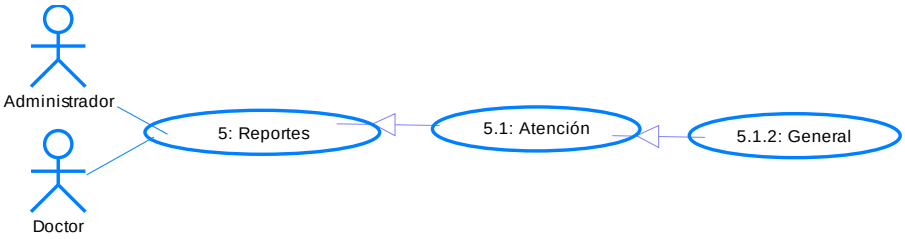
Caso de Uso:	Atención.
Actor:	Administrador , Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden generar reportes de Atención de los pacientes a las citas, es decir saber cuántos pacientes asistieron a las citas.

Escenario 5.1.1



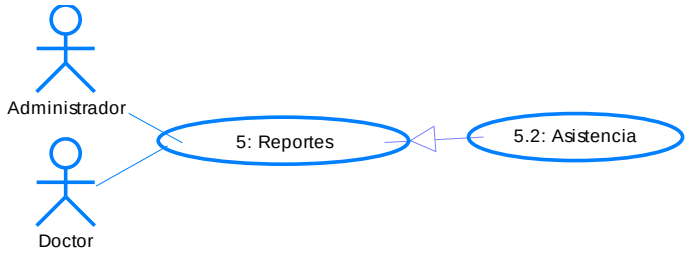
Caso de Uso:	Por Doctor.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden generar reportes de asistencia según el Doctor seleccionado.

Escenario 5.1.2



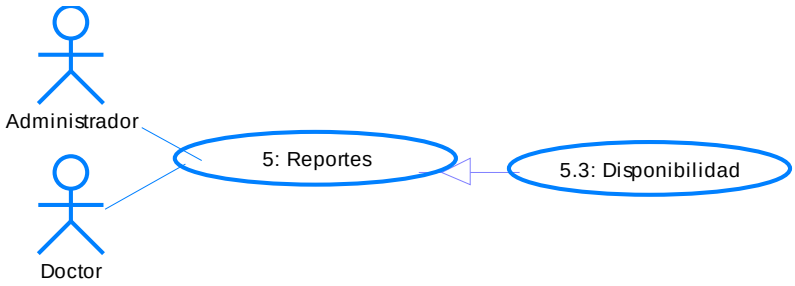
Caso de Uso:	General.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden generar reportes Generales de las atenciones.

Escenario 5.2



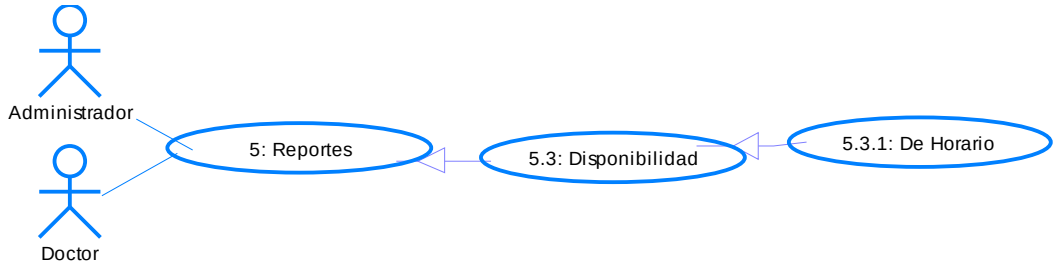
Caso de Uso:	Asistencias.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Primario.
Detalle:	El Administrador y/o Doctor pueden generar reportes sobre las asistencias de los pacientes, para verificar si fueron o no.

Escenario 5.3



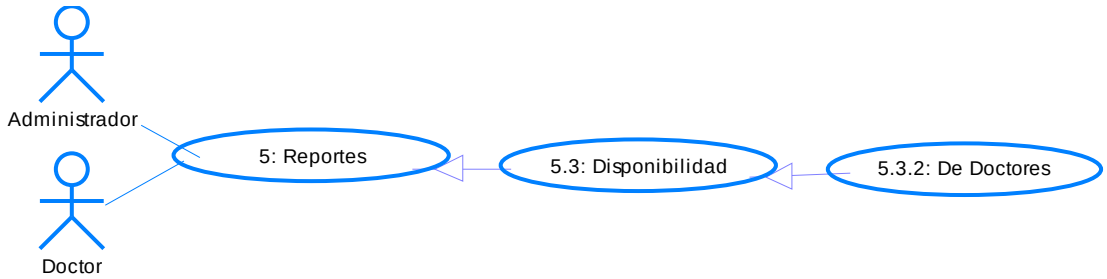
Caso de Uso:	Disponibilidad.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador y/o Doctor pueden generar reportes de Disponibilidad de horarios o doctores para poder asignar las citas.

Escenario 5.3.1



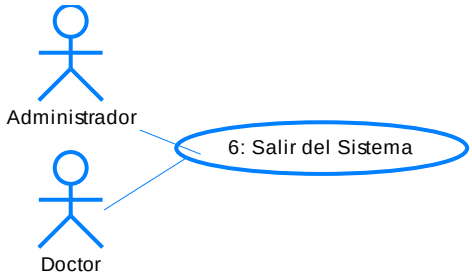
Caso de Uso:	De Horario.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador y/o Doctor pueden generar reportes sobre la disponibilidad de Horarios para asignar citas.

Escenario 5.3.2



Caso de Uso:	De Doctores.
Actor:	Administrador, Doctor
Precondiciones:	Ninguna.
Tipo:	Secundario.
Detalle:	El Administrador y/o Doctor pueden generar reportes sobre la disponibilidad de Doctores para asignar citas.

Escenario 6



Caso de Uso:	Salir del Sistema.
Actor:	Administrador, Doctor.
Precondiciones:	Haber ingresado al Sistema.
Tipo:	Primario.
Detalle:	El Administrador o el Doctor pueden salir del sistema mediante una opción para hacerlo.

4.2 Diagrama de Procesos

Los diagramas de procesos son diseñados con la finalidad de representar los pasos que sigue el algoritmo, es decir nos enseña mediante flechas los procesos que se realizan para hacer una tarea desde el punto inicial hasta el final.

Todo diagrama de procesos tiene un inicio y un fin, teniendo que cada camino que se puede tomar tiene que llegar del principio hasta el término.

Para realizar un diagrama de procesos se debe tener muy en cuenta las ideas principales que deben ser incluidas, para esto en necesario contar con la presencia del dueño o responsable del manejo del sistema, así como los usuarios

La simbología que se usa para realizar este tipo de diagramas es la siguiente:

4.2.1 **Óvalo:** Inicio y término (Abre y/o cierra el diagrama).

4.2.2 **Rectángulo:** Actividad (Representa la ejecución de una o más actividades o procedimientos).

4.2.3 **Rombo:** Decisión (Formula una pregunta o cuestión).

4.2.4 **Círculo:** Conector (Representa el enlace de actividades con otra dentro de un procedimiento).

4.2.5 Triangulo boca abajo: Archivo definitivo (Guarda un documento en forma permanente).

4.2.6 Triangulo boca arriba: Archivo temporal (Proporciona un tiempo para el almacenamiento del documento).

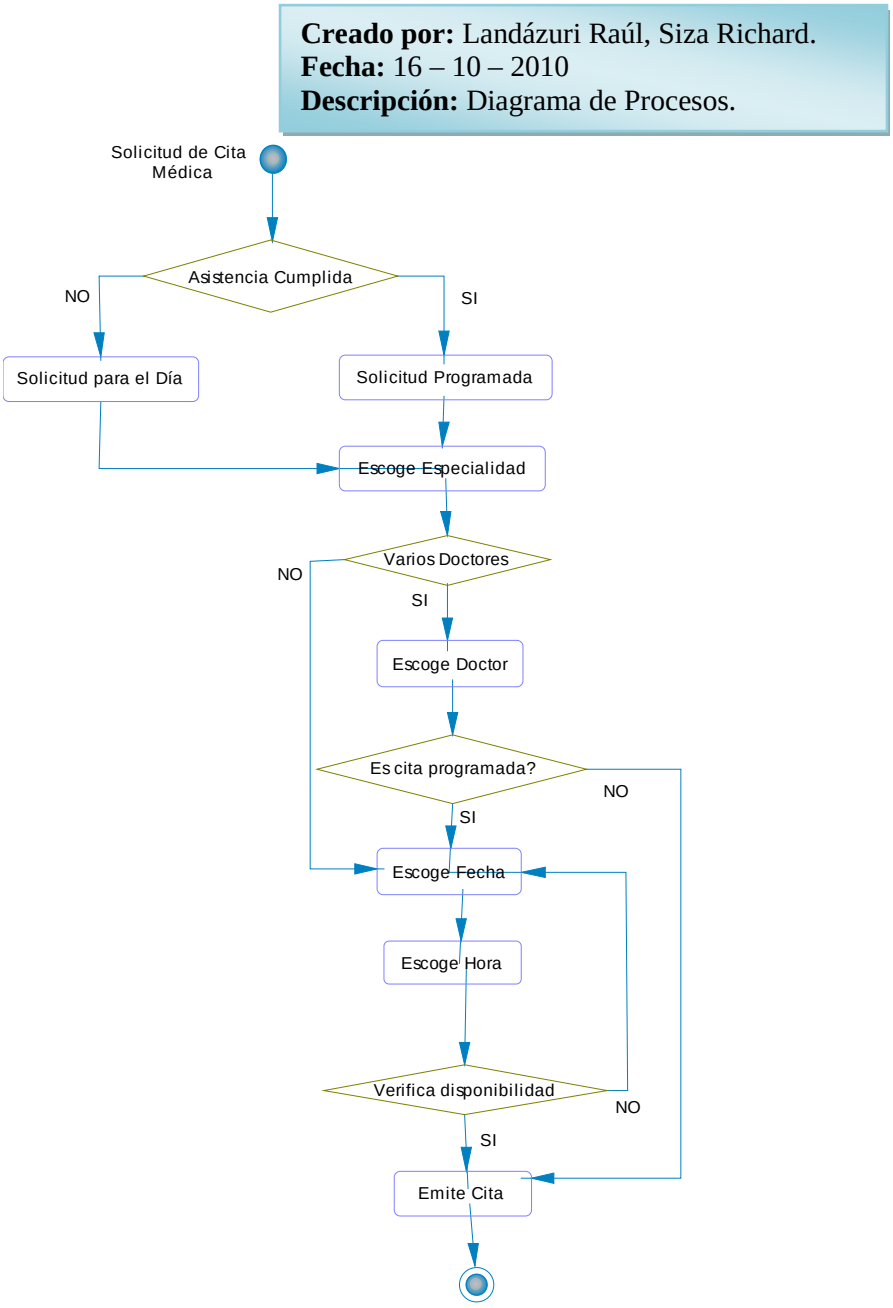


Figura 21
Diagrama de procesos

4.3 Diagrama Entidad – Relación

Creado por: Landázuri Raúl, Siza Richard.
Fecha: 16 – 10 – 2010
Descripción: Diagrama Entidad Relación

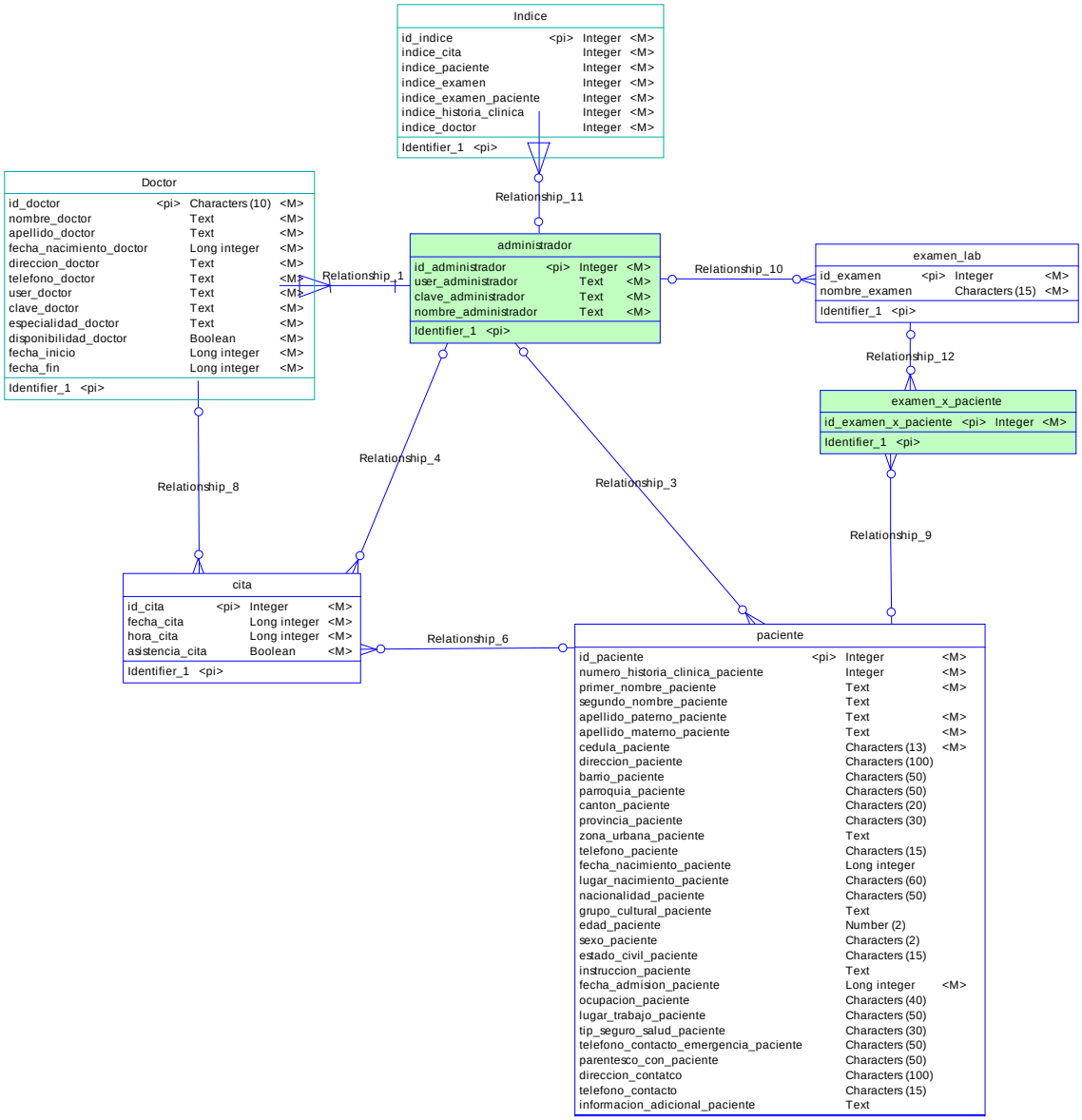


Figura 22
Diagrama entidad relación

CONCLUSIONES

El Sistema de Gestión de Turnos Programados para el Sub Centro de Salud Carapungo que se encuentra instalado y en pleno funcionamiento ha ayudado a disminuir considerablemente la cola de espera en la ventanilla de estadística, debido a la sistematización de la información.

Con la programación extrema se obtuvo mucho mas contacto con el cliente final y por consiguiente se tuvieron bien claros los procesos y requerimientos.

El producto final y los programas que se utilizan para que éste funcione correctamente están enfocados al software libre y siguiendo la misma ideología del actual gobierno, debido a que es para una institución pública.

El trabajo del Sub Centro de Salud con este sistema es más eficiente a comparación con la manera de trabajo anterior, ya que emite un documento impreso con los datos necesarios que el paciente este pendiente.

Los doctores pueden llevar un control más estricto, debido a que tiene en pantalla los pacientes que fueron atendidos y los que serán atendidos.

El sistema puede sacar reportes globales de atención para facilitar al doctor a ver el horario de los pacientes tiene que atender.

RECOMENDACIONES

Se recomienda aumentar un módulo extra para administrar el historial clínico de los pacientes.

Se puede implementar el uso de este sistema mediante internet, para que los pacientes puedan escoger las citas desde cualquier lugar.

Se recomienda establecer un procedimiento para el aviso previo al cumplimiento de una cita, es decir, se alerte al paciente mediante una central telefónica indicando la hora y doctor de la cita.

BIBLIOGRAFIA

http://es.wikipedia.org/wiki/Herramienta_CASE

http://foruminformatika.files.wordpress.com/2007/12/1

http://es.wikipedia.org/wiki/Adobe_Dreamweaver

http://es.wikipedia.org/wiki/XAMPP

http://www.blog-emprendedor.info/la-bpmn-y-su-influencia-en-la-gestion-de-procesos/

http://www.desarrolloweb.com/articulos/840.php

http://www.monografias.com/trabajos51/programacion-extrema/programacion-extrema2.shtml#ciclo

http://www.monografias.com/trabajos51/programacion-extrema/Image3551.gif

http://www.scribd.com/doc/26495149/Programacion-extrema-Informe

http://www.fcad.uner.edu.ar/jai/6JAI/XP_6JAI.pdf

http://books.google.com.ec/books?id=gQWd49zSut4C&pg=PA364&lpg=PA364&dq=ESCENARIOS+DE+PROGRAMACION+EXTREMA&source=bl&ots=s45artAxug&sig=lQ0T6FT-ZngrseRr7QBODHY4nDo&hl=es&ei=16qnS9HzBtOUtgesl_3zAg&sa=X&oi=book_result&ct=result&resnum=3&ved=0CAoQ6AEwAjgK#v=onepage&q=&f=false

http://es.wikipedia.org/wiki/Desarrollo_en_cascada

http://es.wikipedia.org/wiki/Desarrollo_en_espiral

http://es.wikipedia.org/wiki/Modelo_de_prototipos

http://es.wikipedia.org/wiki/Desarrollo_r%C3%A1pido_de_aplicaciones

http://es.wikipedia.org/wiki/Desarrollo_por_etapas

http://www.scribd.com/doc/11446192/Ingenieria-de-Software

http://www.w3c.es/divulgacion/guiasbreves/

http://www.tejedoresdelweb.com/w/CSS

http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada

http://www.programacionphp.net/recursos-manuales/manuales-de-mysql/Manual-oficial-de-w3.org_41.html

http://www.w3c.es/Presentaciones/2005/0314-estandares-JA/

http://mosaic.uoc.edu/ac/le/es/m1/ud2/index.html

http://www.desarrolloweb.com/articulos/186.php

http://es.kioskea.net/contents/css/cssintro.php

http://www.area-ordenadores.com/Metodologias-Web3.html

http://lsi.ugr.es/~gedes/actividades/Dolmen4/a11.pdf

http://www.corgol.com/requisitosparalaweb/index.html

http://www.lifia.info.unlp.edu.ar/papers/2001/Silva2001.pdf

http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_PHP/Hola_mundo

http://geneura.ugr.es/~maribel/php/

http://www.maestrosdelweb.com/editorial/phpintro/

http://www.forosdelweb.com/wiki/Manual_de_PHP:_Qu%C3%A9_es_PHP#Qu.C3.A9_podemos_hacer_con_PHP_y_para_qu.C3.A9_nos_sirve

http://es.wikipedia.org/wiki/ISO/IEC_9126

http://www.lsi.us.es/docencia/get.php?id=2086

http://www.myt.com.pe/principal/Imp_9126.php

http://www.slideshare.net/msch/como-documentar-casos-de-uso

http://www.scribd.com/doc/4060867/Breve-Descripcion-de-Casos-de-Uso

ANEXOS

CERTIFICADO

A petición verbal de interesados señores Raúl Andrés Landázuri Almeida y Richard Vinicio Siza Huilcarema, estudiantes de la Pontificia Universidad Católica del Ecuador, CERTIFICO que solicitaron la realización del Proyecto de Disertación, el cuál consiste en crear un "Sistema de Gestión de Turnos Programados para el Subcentro de Salud Carapungo", el mismo que es un proyecto aplicable para este dispensario, para lo cuál se les prestará todas las facilidades necesarias para la realización del mismo.

Quito, 29 de Diciembre de 2009

Dr. Galo Barragán
Director Área de Salud No 21 (Calderón)

Quito, 09 de diciembre de 2010

CERTIFICADO

Yo Pamela Caicedo, encarda del departamento de estadística del sub centro de salud Carapungo #2 certifico que los señores Raúl Andrés Landázuri Almeida y Richard Vinicio Siza Huilcarema han terminado e instalado el Sistema de Gestión de Turnos Programados para el Sub-centro de Salud, cumpliendo con los requerimiento antes solicitados.

Sírvase hacer uso de la presente para los fines consiguientes.

Atentamente.

Pamela Caicedo

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 1	**Nombre de historia de usuario:** Identificación
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: (Alta)/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio/Bajo)	**Puntos reales:**
Descripción: Identificación e ingreso de usuarios mediante nombre de usuario y contraseña.	
Observaciones: Habrá ciertas restricciones para cada tipo de usuario.	

FIRMA RESPONSABLE

DIRECCIÓN PROVINCIAL DE SALUD DE PICHINCHA
Dra. María T. Aitamirano A.
DIRECTORA
S.C.S Carapungo # 2
ÁREA DE SALUD # 21 CALDERÓN

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 2	**Nombre de historia de usuario:** Administración
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** ?
Prioridad en negocio: (Alta)/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio/Bajo)	**Puntos reales:**
Descripción: Administrar (crear, eliminar, modificar) datos como doctores, areas y usuarios.	
Observaciones: Los datos se verán en una tabla.	

FIRMA RESPONSABLE

DIRECCIÓN PROVINCIAL DE SALUD DE PICHINCHA
Dra. María T. Altamirano A.
DIRECTORA
S.C.S Carapungo # 2
ÁREA DE SALUD # 21 CALDERÓN

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 3	**Nombre de historia de usuario:** Consultas
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** 2
Prioridad en negocio: (Alta/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio/Bajo)	**Puntos reales:**
Descripción: Se podrán consultar datos de doctores, areas, usuarios y citas, mediante parámetros especificos y/o generales.	
Observaciones:	

FIRMA RESPONSABLE

DIRECCION PROVINCIAL DE SALUD DE PICHINCHA
Dra. Maria T. Altamirano A.
DIRECTORA
S.C.S Carapungo # 2
AREA DE SALUD # 31 CALDERON

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 4	**Nombre de historia de usuario:** Crear citas
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** 2
Prioridad en negocio: (Alta)/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto)/Medio/Bajo)	**Puntos reales:**
Descripción: Se podrán crear nuevas citas a partir de horarios disponibles de los doctores de 7 am a 4:30 pm	
Observaciones:	

FIRMA RESPONSABLE

Dra. María T. Altamirano A. DIRECTORA

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 5	**Nombre de historia de usuario:** Impresiones citas
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** ?
Prioridad en negocio: (Alta/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio/Bajo)	**Puntos reales:**
Descripción: Se podrá imprimir la cita para que quede constancia de la misma y sea entregado al paciente.	
Observaciones:	

FIRMA RESPONSABLE

DIRECCION PROVINCIAL DE SALUD DE PICHINCHA
Dra. Maria T. Altamirano A.
DIRECTORA
S.C.S Carapungo # 2
AREA DE SALUD # 21 CALDERON

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 6	**Nombre de historia de usuario:** Impresiones lab.
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: (Alta/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio/Bajo)	**Puntos reales:**
Descripción: Se podrá imprimir datos especificos de laboratorio	
Observaciones:	

DIRECCION PROVINCIAL DE SALUD PICHINCHA
Dra. María T. Altamirano A.
DIRECTORA
SECRETARIA
AREA DE SALUD N° 2

Apellido Casilla

FIRMA RESPONSABLE

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

DISEÑO, DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE GESTIÓN DE TURNOS PROGRAMADOS PARA EL SUB CENTRO DE SALUD CARAPUNGO

HISTORIAS DE USUARIO

HISTORIA DE USUARIO	
Numero: 7	**Nombre de historia de usuario:** Disponibilidad
Modificación/extensión de historia de usuario (Nª y Nombre):	
Usuario:	**Iteración asignada:** 1
Prioridad en negocio: (Alta)/Media/Baja)	**Puntos estimados:**
Riesgo en desarrollo: (Alto/Medio)/Bajo)	**Puntos reales:**
Descripción: Se controlará la disponibilidad de tiempo de cada médico en su especialidad para poder restringir la entrega de citas.	
Observaciones:	

FIRMA RESPONSABLE

Richard Vinicio Siza H.
Raúl Andrés Landázuri A.

Tabla de contenido