



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA  
COMPUTACIÓN**

**CARRERA DE TECNOLOGÍA EN SISTEMAS DE  
INFORMACIÓN**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN  
DEL TÍTULO DE TECNOLOGÍA EN SISTEMAS DE  
INFORMACIÓN**

**AUTOR: ANDRES GABRIEL PERALTA TOMALO**

**DESARROLLO E IMPLEMENTACION DE UNA  
APLICACIÓN ORIENTADA A LA WEB PARA EL CONTROL  
DE LA EJECUCIÓN DE PROYECTOS EN LA EMPRESA  
INCOAYAM**

**DIRECTOR: ING. GILMA TOAZA**

**CODIRECTOR: ING. GEOVANNI NINAHUALPA**

**SALGOLQUÍ**

**2015**

**CERTIFICADO**

Ing. Gilma Toaza

Ing. Geovanni Ninahualpa

**CERTIFICAN**

Que el trabajo titulado **DESARROLLO E IMPLEMENTACION DE UNA APLICACIÓN ORIENTADA A LA WEB PARA EL CONTROL DE LA EJECUCIÓN DE PROYECTOS EN LA EMPRESA INCOAYAM**, realizado por Andrés Gabriel Peralta, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de Las Fuerzas Armadas ESPE.

Debido al interés de su contenido recomiendan su publicación. El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (PDF) y en documento Word.

Autorizan a Andrés Gabriel Peralta Tomalo que lo entregue al Ing. Fernando Galarraga, en su calidad de Director de Carrera.

Sangolquí, Abril 2015

**Ing. Gilma Toaza****DIRECTOR****Ing. Geovanni Ninahualpa****CODIRECTO**

## DECLARACION DE RESPONSABILIDAD

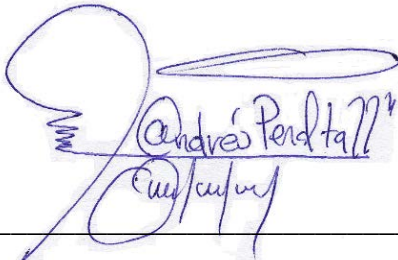
Yo: PERALTA TOMALO ANDRES GABRIEL

### DECLARO QUE:

El proyecto de grado denominado “DESARROLLO E IMPLEMENTACION DE UNA APLICACIÓN ORIENTADA A LA WEB PARA EL CONTROL DE LA EJECUCIÓN DE PROYECTOS EN LA EMPRESA INCOAYAM”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la lista de referencias. Consecuentemente este trabajo es de mi autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, Abril 2015



Handwritten signature of Andrés Peralta in blue ink, written over a horizontal line. The signature includes the name 'Andrés Peralta' and a date '2015'.

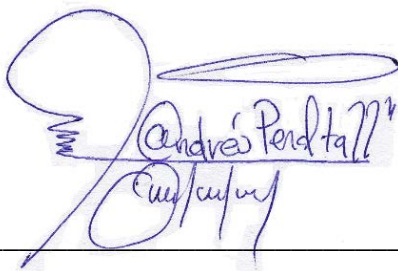
**Peralta Tomalo Andrés Gabriel**

## AUTORIZACIÓN

Yo, Peralta Tomalo Andrés Gabriel

Autorizo a la Universidad de las Fuerzas Armadas ESPE la publicación, en la biblioteca virtual de la Institución del trabajo DESARROLLO E IMPLEMENTACION DE UNA APLICACIÓN ORIENTADA A LA WEB PARA EL CONTROL DE LA EJECUCIÓN DE PROYECTOS EN LA EMPRESA INCOAYAM, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Sangolquí, Abril 2015



Handwritten signature of Andrés Peralta in blue ink, written over a horizontal line. The signature includes the name 'Andrés Peralta' and a stylized initial or mark below it.

**Peralta Tomalo Andrés Gabriel**

## **DEDICATORIA**

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional.

A mi madre, por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional sin importar nuestras diferencias de opiniones.

A mi padre, que me ha sabido formarme con buenos sentimientos, hábitos y valores, lo cual me ha ayudado a salir adelante en los momentos más difíciles.

A mis hermanos Katherine, Nicole, Josué, que son una parte importante en mi vida y que me han brindado la oportunidad de ser una guía en su vida.

A mi familia en general, porque me han brindado su apoyo incondicional y por compartir conmigo buenos y malos momentos.

## **AGRADECIMIENTO**

A Dios, por regalarme la sabiduría para seguir adelante todos los días. A mi Mami Patricia quien más que una buena madre ha sido mi ayuda incondicional durante toda mi Carrera, me ha consentido y apoyado en lo que me he propuesto y sobre todo ha sabido corregir mis errores.

Agradezco también a mi Padre Manuel por ser en apoyo en mi Carrera, en mis logros, por brindarme esos consejos que solo buscan lo mejor para mi vida.

A mis hermanos por todo el amor que han sabido entregarme, durante todo el transcurso de mi meta.

A mis profesores, gracias por su tiempo, por su apoyo así como por la sabiduría que me transmitieron en el desarrollo de mi formación profesional.

## INDICE DE CONTENIDOS

CERTIFICADO .....	ii
DECLARACION DE RESPONSABILIDAD .....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA .....	v
AGRADECIMIENTO.....	vi
RESUMEN.....	xvi
ABSTRACT.....	xvii
Capítulo I.....	1
1 MARCO TEORICO .....	1
1.1    Introducción a XP (Programación Extrema).....	1
1.1.1    Objetivos de la programación extrema.....	1
1.1.2    Fases de la programación extrema. ....	4
1.2    Descripción del negocio.....	7
1.3    PHP .....	8
1.3.1    Historia.....	8
1.3.2    ¿Qué es?.....	9
1.3.3    ¿Para qué sirve?.....	9
1.3.4    ¿Cómo trabaja PHP? .....	10
1.3.5    ¿Qué ocurre? .....	10
1.3.6    Sintaxis.....	10
1.3.7    Ventajas.....	11
1.3.8    Desventajas. ....	13
1.4    JavaScript (abreviado comúnmente "JS").....	13
1.4.1    Historia.....	14
1.4.2    Ventajas.....	15
1.4.3    Desventajas. ....	16
1.4.4    Sintaxis y Semántica. ....	17
1.5    Netbeans.....	18
1.5.1    Historia.....	19
1.5.2    PHP + NetBeans.....	19
1.5.3    Características. ....	19
1.6    HTML HyperText Markup Language (Lenguaje de Marcado de Hipertexto).....	21
1.6.1    Historia.....	21
1.6.2    Estructura. ....	21
1.6.3    Ventajas.....	23
1.6.4    Hojas de estilo (CSS). ....	23
1.7    AJAX Asynchronous JavaScript y XML.....	25

1.7.1	Historia.....	25
1.7.2	Ventajas.....	26
1.7.3	Desventajas. ....	26
1.8	MySQL.....	27
1.8.1	Lenguajes de programación.....	27
1.8.2	Versiones.....	28
1.8.3	Especificaciones. ....	28
1.8.4	Licencia. ....	28
1.9	Apache. ....	30
1.9.1	Historia.....	31
1.9.2	Ventajas.....	32
<b>Capítulo II.....</b>		<b>35</b>
<b>2</b>	<b>PLANEACION.....</b>	<b>35</b>
2.1	Historias de Usuario.....	35
2.1.1	Módulos del Sistema: .....	36
2.1.2	Especificación de historias de usuario por módulos del sistema .....	37
2.1.3	Valoración de historias de usuario .....	48
2.2	Plan de Entrega .....	49
2.2.1	Tiempo Calendario.....	49
2.2.2	Esfuerzo de Desarrollo.....	50
2.2.3	Elaboración del Plan de Entrega .....	51
2.3	Cuadro de entregables.....	51
2.3.1	Entregables en XP.....	51
2.4	Planificación de iteraciones. ....	56
2.4.1	Tareas de ingeniería. ....	57
<b>Capítulo III.....</b>		<b>62</b>
<b>3</b>	<b>DISEÑO.....</b>	<b>62</b>
3.1	Diseño simple.....	62
3.2	Metáforas .....	62
3.3	Refactorización .....	63
3.4	No agregar funcionalidades antes de lo planeado .....	63
3.4.1	Especificación de la arquitectura del sistema .....	64
3.5	Tarjetas de Responsabilidad Colaboración (CRC).....	65
3.5.1	Tarjetas CRC primera iteración.....	66
3.5.2	Tarjetas CRC segunda iteración. ....	69
3.5.3	Tarjetas CRC tercera iteración. ....	71
3.5.4	Tarjetas CRC cuarta iteración. ....	73
3.5.5	Tarjetas CRC quinta iteración. ....	75
3.6	Diagrama Entidad Relación .....	78
<b>4</b>	<b>CODIFICACION.....</b>	<b>79</b>



4.1	Cliente - Programador.....	79
4.2	Programación en pareja.....	79
4.3	Integración secuencial.....	79
4.4	Integraciones frecuentes.....	80
4.5	Estándares de programación .....	80
4.6	Desarrollo de la aplicación.....	81
4.6.1	Página principal.....	81
4.6.2	Archivo común ajax .....	83
4.6.3	Archivo server .....	84
<b>5</b>	<b>PRUEBAS .....</b>	<b>86</b>
5.1.1	Pruebas de aceptación primera iteración. ....	86
5.1.2	Pruebas de aceptación segunda iteración.....	102
5.1.3	Pruebas de aceptación tercera iteración.....	107
5.1.4	Pruebas de aceptación cuarta iteración.....	111
5.1.5	Pruebas de aceptación quinta iteración.....	121
<b>6</b>	<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>129</b>
6.1	Conclusiones .....	129
6.2	Recomendaciones .....	130
	<b>BIBLIOGRAFIA.....</b>	<b>132</b>

## INDICE DE TABLAS

<b>Tabla 1.</b> Estimación de tiempo de historias de usuarios .....	49
<b>Tabla 2.</b> Tabla tiempo calendario.....	50
<b>Tabla 3.</b> Esfuerzo de desarrollo en base a una persona.....	50
<b>Tabla 4.</b> Historial de versiones por historia de usuario.....	52
<b>Tabla 5.</b> Historial de seguimiento de CRC.....	52
<b>Tabla 6.</b> Historial de Seguimiento de la Ejecución de Pruebas de Aceptación.....	53
<b>Tabla 7.</b> Historial de seguimiento de iteraciones .....	55
<b>Tabla 8.</b> Historial de seguimiento de entrega final.....	56

## ÍNDICE DE FIGURAS

Figura 1. Tutorial básico PHP .....	10
Figura 2. Ejemplo básico PHP .....	11
Figura 3. Funcionamiento de javascript .....	14
Figura 4. Código Javascript .....	18
Figura 5. Estructura HTML.....	22
Figura 6. Esquema AJAX .....	25
Figura 7. Plantilla de historias de usuario .....	36
Figura 8. Historia de usuario registro de usuarios .....	38
Figura 9. Historia de usuario prioridades de proyectos.....	38
Figura 10. Historia de usuario perfil de usuarios .....	39
Figura 11. Historia de usuario líneas de negocio .....	39
Figura 12. Historia de usuario perfil de trabajador .....	40
Figura 13. Historia de usuario estados de proyectos.....	40
Figura 14. Historia de usuario contactos.....	41
Figura 15. Historia de usuario ingreso cabecera kit.....	41
Figura 16. Historia de usuario ingreso materia prima.....	42
Figura 17. Historia de usuario registro de mano de obra .....	42
Figura 18. Historia de usuario crear una oferta.....	43
Figura 19. Historia de usuario ingreso de estados a la oferta .....	43
Figura 20. Historia de usuario ingreso de contactos a la oferta .....	44
Figura 21. Historia de usuario cabecera de la cotización.....	44
Figura 22. Historia de usuario kits a la cotización .....	45
Figura 23. Historia de usuario gastos a la cotización.....	45
Figura 24. Historia de usuario datos informativos a la cotización.....	46
Figura 25. Historia de usuario datos al proyecto .....	46
Figura 26. Historia de usuario personas involucradas al proyecto.....	47
Figura 27. Historia de usuario clientes al proyecto.....	47
Figura 28. Historia de usuario otros datos al proyecto.....	48
Figura 29. Historia de usuario tareas al proyecto.....	48
Figura 30. Planificación de iteración por historia de usuario .....	59
Figura 31. Planificación de la primera iteración .....	59
Figura 32. Planificación de la segunda iteración .....	60
Figura 33. Planificación de la tercera iteración.....	60
Figura 34. Planificación de la cuarta iteración.....	61

Figura 35. Planificación de la quinta iteración.....	61
Figura 36. Esquema n-capas .....	65
Figura 37. Plantilla base tarjeta CRC.....	66
Figura 38. Tarjeta CRC_usuarios.....	66
Figura 39. Tarjeta CRC_prioridades_de_proyecto .....	67
Figura 40. Tarjeta CRC_perfiles_de_usuario .....	67
Figura 41. Tarjeta CRC_lineas_de_negocio .....	68
Figura 42. Tarjeta CRC_perfiles_de_trabajador .....	68
Figura 43. Tarjeta CRC_estados_de_proyecto .....	69
Figura 44. Tarjeta CRC_contactos .....	69
Figura 45. Tarjeta CRC_cabeceras_de_kit .....	70
Figura 46. Tarjeta CRC_materia_prima_en_kit.....	70
Figura 47. Tarjeta CRC_mano_de_obra_en_kit .....	71
Figura 48. Tarjeta CRC_ofertas .....	72
Figura 49. Tarjeta CRC_estados_en_oferta .....	72
Figura 50. Tarjeta CRC_contactos_en_oferta.....	73
Figura 51. Tarjeta CRC_cabeceras_de_cotizacion .....	73
Figura 52. Tarjeta CRC_kits_en_la_cotizacion .....	74
Figura 53. Tarjeta CRC_otros_gastos_en_la_cotizacion.....	74
Figura 54. Tarjeta CRC_datos_informativos_en_la_cotizacion .....	75
Figura 55. Tarjeta CRC_cabeceras_de_proyectos .....	75
Figura 56. Tarjeta CRC_personas_involucradas_en_proyecto.....	76
Figura 57. Tarjeta CRC_clientes_en_proyecto .....	76
Figura 58. Tarjeta CRC_otros_datos_en_proyecto.....	77
Figura 59. Tarjeta CRC_tareas_al_proyecto.....	77
Figura 60. Diagrama entidad relación.....	78
Figura 61. Cabecera página principal.....	81
Figura 62. Bloque de scripts página principal.....	82
Figura 63. Estructura HTML página principal.....	82
Figura 64. Footer página principal .....	83
Figura 65. Librerías archivo común .....	84
Figura 66. Instancia ajax archivo común .....	84
Figura 67. Registro funciones archivo común .....	84
Figura 68. Require del archivo server .....	85
Figura 69. Funciones del archivo server .....	85
Figura 70. Request del archivo server.....	85

Figura 71. Plantilla pruebas de aceptación.....	86
Figura 72. Prueba de aceptación guardar usuario .....	88
Figura 73. Prueba de aceptación buscar usuario .....	88
Figura 74. Prueba de aceptación editar usuario .....	89
Figura 75. Prueba de aceptación eliminar usuario .....	89
Figura 76. Prueba de aceptación resetear clave .....	90
Figura 77. Prueba de aceptación guardar prioridad de proyecto.....	90
Figura 78. Prueba de aceptación buscar prioridad de proyecto .....	91
Figura 79. Prueba de aceptación editar prioridad de proyecto.....	91
Figura 80. Prueba de aceptación eliminar prioridad de proyecto.....	92
Figura 81. Prueba de aceptación guardar perfil .....	92
Figura 82. Prueba de aceptación buscar perfil .....	93
Figura 83. Prueba de aceptación editar perfil.....	93
Figura 84. Prueba de aceptación eliminar perfil .....	94
Figura 85. Prueba de aceptación guardar línea .....	94
Figura 86. Prueba de aceptación buscar línea .....	95
Figura 87. Prueba de aceptación editar línea .....	95
Figura 88. Prueba de aceptación eliminar línea .....	96
Figura 89. Prueba de aceptación guardar perfil de trabajador .....	96
Figura 90. Prueba de aceptación buscar perfil trabajador .....	97
Figura 91. Prueba de aceptación editar perfil trabajador .....	97
Figura 92. Prueba de aceptación eliminar perfil de trabajador .....	98
Figura 93. Prueba de aceptación guardar estado .....	98
Figura 94. Prueba de aceptación buscar estado.....	99
Figura 95. Prueba de aceptación editar estado .....	99
Figura 96. Prueba de aceptación eliminar estado.....	100
Figura 97. Prueba de aceptación guardar contacto.....	100
Figura 98. Prueba de aceptación buscar contacto .....	101
Figura 99. Prueba de aceptación editar contacto.....	101
Figura 100. Prueba de aceptación eliminar contacto .....	102
Figura 101. Prueba de aceptación guardar kit.....	103
Figura 102. Prueba de aceptación búsqueda kit.....	103
Figura 103. Prueba de aceptación editar kit.....	104
Figura 104. Prueba de aceptación eliminar kit.....	104
Figura 105. Prueba de aceptación agregar detalle materia prima .....	105
Figura 106. Prueba de aceptación editar detalle materia prima .....	105

Figura 107. Prueba de aceptación eliminar detalle materia prima.....	106
Figura 108. Prueba de aceptación agregar detalle mano de obra.....	106
Figura 109. Prueba de aceptación editar detalle mano de obra.....	107
Figura 110. Prueba de aceptación eliminar detalle mano de obra.....	107
Figura 111. Prueba de aceptación guardar oferta.....	108
Figura 112. Prueba de aceptación buscar oferta.....	108
Figura 113. Prueba de aceptación editar oferta.....	109
Figura 114. Prueba de aceptación agregar estado a la oferta.....	109
Figura 115. Prueba de aceptación eliminar estado de la oferta.....	110
Figura 116. Prueba de aceptación agregar contacto a la oferta.....	110
Figura 117. Prueba de aceptación crear contacto.....	111
Figura 118. Prueba de aceptación eliminar contacto de oferta.....	111
Figura 119. Prueba de aceptación guardar cotización.....	112
Figura 120. Prueba de aceptación buscar cotización.....	112
Figura 121. Prueba de aceptación editar cotización.....	113
Figura 122. Prueba de aceptación copiar cotización.....	113
Figura 123. Prueba de aceptación calcular cotización.....	114
Figura 124. Prueba de aceptación agregar kit.....	114
Figura 125. Prueba de aceptación consultar kit.....	115
Figura 126. Prueba de aceptación editar kit.....	115
Figura 127. Prueba de aceptación eliminar kit.....	115
Figura 128. Prueba de aceptación calcular kit.....	116
Figura 129. Prueba de aceptación editar materia prima.....	116
Figura 130. Prueba de aceptación editar mano de obra.....	117
Figura 131. Prueba de aceptación agregar gasto aumenta cotización.....	117
Figura 132. Prueba de aceptación eliminar gasto aumenta cotización.....	118
Figura 133. Prueba de aceptación agregar gasto recalcula cotización.....	118
Figura 134. Prueba de aceptación eliminar gasto recalcula cotización.....	119
Figura 135. Prueba de aceptación agregar gasto extras cotización.....	119
Figura 136. Prueba de aceptación eliminar gasto extras cotización.....	120
Figura 137. Prueba de aceptación agregar dato informativo.....	120
Figura 138. Prueba de aceptación eliminar dato informativo.....	120
Figura 139. Prueba de aceptación guardar proyecto.....	121
Figura 140. Prueba de aceptación buscar proyecto.....	122
Figura 141. Prueba de aceptación editar proyecto.....	122
Figura 142. Prueba de aceptación agregar personas involucradas.....	123

Figura 143. Prueba de aceptación eliminar personas involucradas.....	123
Figura 144. Prueba de aceptación agregar cliente.....	124
Figura 145. Prueba de aceptación eliminar cliente .....	124
Figura 146. Prueba de aceptación agregar otro dato .....	125
Figura 147. Prueba de aceptación eliminar otro dato.....	125
Figura 148. Prueba de aceptación guardar tarea .....	126
Figura 149. Prueba de aceptación editar tarea .....	126
Figura 150. Prueba de aceptación eliminar tarea .....	127
Figura 151. Prueba de aceptación agregar kit a la tarea.....	127
Figura 152. Prueba de aceptación eliminar kit de tarea .....	127
Figura 153. Prueba de aceptación agregar responsable a la tarea .....	128
Figura 154. Prueba de aceptación eliminar responsable de tarea.....	128

## **RESUMEN**

El presente proyecto de tesis es una aplicación de los diversos conocimientos adquiridos durante la Carrera de Tecnología en Sistemas de Información. El objetivo fue desarrollar una solución para controlar la gestión de proyectos de la empresa INCOAYAM, la empresa posee un ERP. La presente tesis detalla en el capítulo I todo lo teórico acerca de la metodología XP, así como conceptos básicos acerca de las herramientas que se va a usar durante el desarrollo del problema, el capítulo II se documenta todos los pasos referentes a la planificación del sistema, en el capítulo III se realiza el diseño con todas las recomendaciones que XP aconseja, posteriormente el capítulo IV cubre en sí la codificación de la aplicación utilizando la metodología XP, el capítulo V documenta todas las pruebas que el sistema tiene que pasar para poder ser implantado. Finalizando en el capítulo VI con las conclusiones y recomendaciones.

### **Palabras clave:**

- **METODOLOGIA XP**
- **ERP**
- **DISEÑO**
- **CODIFICACION**
- **PRUEBAS**



## **ABSTRACT**

This thesis project is an application of the various knowledge acquired during the Career Technology in Information Systems. The purpose was to develop a solution for controlling project management company INCOAYAM, the company has an ERP. This thesis detailed in Chapter I all theoretical about the XP methodology and basic concepts about the tools that will be used during the development of the problem, chapter II all steps concerning the planning system is documented in Chapter III design is done with all recommendations XP advised subsequently chapter IV covers itself coding the application using the XP methodology, chapter V documents all the evidence that the system has to go to be implanted. Finishing in Chapter VI the conclusions and recommendations.

### **Keywords:**

- **METHODOLOGY**
- **ERP**
- **DESIGN**
- **CODING**
- **TESTS**

## CAPÍTULO I

### 1 MARCO TEORICO

#### 1.1 Introducción a XP (Programación Extrema)

La programación extrema (XP) es un enfoque de la ingeniería de software formulado por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software. Una de las diferencias que posee la programación extrema sobre las demás metodologías tradicionales principalmente es que pone mucho más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha de un proyecto son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que el poder adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en el mantenimiento sobre los requisitos posteriores (Universidad Unión Bolivariana, 2010).

##### 1.1.1 Objetivos de la programación extrema.

El objetivo principal de la XP es la satisfacción del cliente. Se le trata de dar al cliente lo que quiere y cuando quiere. Por consecuente, se debe responder de la manera más rápida y eficiente a las necesidades del cliente, aunque este realice cambios en fases que el proyecto ya tenga un cierto avance. Por ser considerada una metodología Ágil, se pueden producir modificaciones de los requisitos del proyecto a lo largo de su desarrollo, sin que esto produzca un buen dolor de cabeza. Otro de los objetivos es que maneja la metodología es el trabajo en grupo. Por este motivo los jefes del proyecto, clientes y desarrolladores forman parte del equipo y deben estar involucrados en el desarrollo del mismo (Valverde, 2007).

##### Valores de XP.

La Programación Extrema “Extreme Programming” (XP) no es un conjunto de reglas a seguir, sino una forma de trabajar en armonía con los valores personales y organizacionales, que tiene su punto de partida en cinco valores fundamentales que son: Comunicación, Simplicidad, Retroalimentación, Coraje y Respeto.

Los valores XP representan un excelente punto de partida para entender los cambios de paradigmas que implica trabajar bajo la filosofía ágil.

- **Comunicación:** “Everyone is part of the team and we communicate face to face daily. We will work together on everything from requirements to code. We will create the best solution to our problem that we can together” (Wells, 2009).

En los métodos tradicionales, la comunicación de los requerimientos del proyecto entre desarrolladores se realiza a través de documentación, un ejemplo concreto es Rational Unified Process. XP rompe con este esquema, la comunicación se realiza por medio de transferencia de conocimientos en reuniones frecuentes cara a cara entre los usuarios y desarrolladores, lo que le da a ambos una visión compartida del sistema. Por ello, XP favorece diseños simples, colaboración de usuarios con programadores, comunicación verbal frecuente, retroalimentación y construcción rápida del software (PMOinformatica, 2012).

- **Simplicidad:** “We will do what is needed and asked for, but no more. This will maximize the value created for the investment made to date. We will take small simple steps to our goal and mitigate failures as they happen. We will create something we are proud of and maintain it long term for reasonable costs” (Wells, 2009).

En XP se comienza desarrollando las soluciones más sencillas necesarias para solucionar los problemas que se están viendo en ese momento, añadiendo funcionalidad extra más tarde, en la medida en que se obtiene más información de los requerimientos. La diferencia respecto a esquemas

tradicionales es que se enfoca en las necesidades de hoy en lugar de las necesidades de mañana, la semana que viene o el mes que viene.

La ventaja es que no se invierte esfuerzo en futuros requerimientos que podrían cambiar o que no se vayan a necesitar. Asimismo, un diseño y programación simple mejora la calidad de las comunicaciones, pues es más fácil de implementar y entender por todos en el equipo (PMOinformatica, 2012).

- **Retroalimentación:** “We will take every iteration commitment seriously by delivering working software. We demonstrate our software early and often then listen carefully and make any changes needed. We will talk about the project and adapt our process to it, not the other way around”(Wells, 2009).

Al estar el cliente integrado en el Proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en lo que es más importante. Considérense los problemas que derivan de tener ciclos muy largos. Meses de trabajo pueden tirarse por la borda debido a cambios en los criterios del cliente o malentendidos por parte del equipo de desarrollo. El código también es una fuente de retroalimentación gracias a las Herramientas de desarrollo. Por ejemplo, las Pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código(PMOinformatica, 2012).

- **Coraje:** “We will tell the truth about progress and estimates. We don't document excuses for failure because we plan to succeed. We don't fear anything because no one ever works alone. We will adapt to changes when ever they happen”(Wells, 2009).

Algunas prácticas del coraje están relacionadas a diseñar y programar para hoy y no para mañana, evitando así hacer énfasis en el diseño en detrimento de todo lo demás. Refactorizar el código siempre que sea necesario sin titubear al respecto, para ello se inspeccionara constantemente el código y modificara, de tal manera que futuros cambios se puedan implementar más fácilmente, desarrollar rápido para atender las necesidades de hoy pero refactorizar después para facilitar el mantenimiento. Desechar componentes o piezas de código cuando sea necesario, sin preocuparse del tiempo invertido y perdido en su creación ya que es mejor desechar algo que no es útil en lugar de tratar de repararlo (PMOinformatica, 2012).

- **Respeto:** “Everyone gives and feels the respect they deserve as a valued team member. Everyone contributes value even if it's simply enthusiasm. Developers respect the expertise of the customers and vice versa. Management respects our right to accept responsibility and receive authority over our own work” (Wells, 2009).

Respeto es tanto por el trabajo de los demás como por el trabajo de uno mismo, por ejemplo, los desarrolladores nunca deben subir cambios que impidan la compilación de la versión, que hagan fallar pruebas unitarias ya realizadas o que de alguna otra forma retrasen el trabajo de sus pares, esto significa tener respeto por el trabajo y el tiempo de los demás.

Asimismo, los desarrolladores respetan su propio trabajo por medio de su compromiso con una alta calidad y buscando el mejor diseño para la solución por medio de la refactorización constante. En cuanto al trabajo en equipo, nadie debe sentirse poco apreciado o ignorado, todos deben colaborar en esto, tratando con respeto a sus compañeros y mostrando respeto por sus opiniones, esto asegura altos niveles de motivación y lealtad hacia el proyecto (PMOinformatica, 2012).

### 1.1.2 Fases de la programación extrema.

**Planeación.**

En este punto se tendrá que elaborar la planificación por etapas, donde se aplicarán diferentes iteraciones. Para hacerlo será necesaria la existencia de reglas que se han de seguir por las partes implicadas en el proyecto para que todas las partes tengan voz y se sientan realmente partícipes de la decisión tomada. Las entregas se tienen que hacer cuanto antes mejor, y con cada iteración, el cliente ha de recibir una nueva versión. Cuanto más tiempo se tarde en introducir una parte esencial, menos tiempo se tendrá para trabajar con ella después. Se aconseja muchas entregas y muy frecuentes. De esta manera un error en la parte inicial del sistema tiene más posibilidades de detectarse rápidamente. Una de las máximas a aplicar es, los cambios, no han de suponer más horas de programación para el programador, ya que el que no se termina en un día, se deja para el día siguiente.

Se ha de tener asumido que en el proceso de planificación habrán errores, es más, serán comunes, y por esto esta metodología ya los tiene previstos, por lo tanto se establecerán mecanismos de revisión. Cada tres o cinco iteraciones es normal revisar las historias de los usuarios, y renegociar la planificación. Cada iteración necesita también ser planificada, es lo que se llama planificación iterativa, en la que se anotarán las historias de usuarios que se consideren esenciales y las que no han pasado las pruebas de aceptación. Estas planificaciones también se harán en tarjetas, en las que se escribirán los trabajos que durarán entre uno y tres días (Anca & Borda, 2014).

Es por esto que el diseño se puede clasificar como continuo. Añade agilidad al proceso de desarrollo y evita que se mire demasiado hacia delante, desarrollando trabajos que aún no han estado programados. Este tipo de planificación en iteraciones y el diseño iterativo, hace que aparezca una práctica que no existía en la programación tradicional. Se trata de las discusiones diarias informales, para fomentar la comunicación, y para hacer que los desarrolladores tengan tiempo de hablar de los problemas a los que se enfrentan y de ver cómo van con sus trabajos.

## **Diseño.**

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

Tenemos que codificar porque sin código no hay programas, tenemos que hacer pruebas porque sin pruebas no sabemos si hemos acabado de codificar, tenemos que escuchar, porque si no escuchamos no sabemos qué codificar ni probar, y tenemos que diseñar para poder Codificar, probar y escuchar indefinidamente(Delgado, 2008).

Las características fundamentales del método son:

- **Desarrollo iterativo e incremental:** pequeñas mejoras, unas tras otras.
- **Pruebas unitarias:**Estas tienen que ser continuas, frecuentemente repetidas y automatizadas, incluyendo Pruebas de Regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- **Desarrollo:**Se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto, pero para este desarrollo se lo hará en una sola persona.
- **Cliente-Desarrollador:**Frecuente integración del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Entregas:**Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

- **Refactorización del código:** Es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- **Propiedad del código compartida:** En vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.
- **Simplicidad en el código:** Cuando los niveles más complejos del desarrollo estén ya realizados se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La Simplicidad con la Comunicación dentro del desarrollo son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Cuanto más simple es el sistema, menos tendrá que comunicar sobre éste, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores (Marquetti Asociados, 2010).

## 1.2 Descripción del negocio

INCOAYAM es una compañía de Ingeniería y Construcciones especializada en diseñar y construir redes de distribución de gases combustibles. Nació en 1999 e inmediatamente se asoció comercialmente con la multinacional REPSOL quienes comercializan gas GLP en el Ecuador a través de DURAGAS S.A. con su producto REPSOLGAS.

INCOAYAM ha desarrollado exitosamente el mercado de instalaciones de gas GLP en el Ecuador, posicionándose como líder del sector en el Ecuador.



La empresa INCOAYAM dispone de un ERP cliente servidor llamado JIREH que controla todo el aspecto financiero, comercial de la empresa, la necesidad de llevar un control exacto de la vida de los proyectos en ejecución ha obligado a la empresa a valerse de la herramienta de ofimática Microsoft Excel.

La empresa trabajo por algún tiempo de manera eficaz creando hojas de cálculo con sus respectivas celdas relacionadas dentro de un mismo libro de Excel así como muchos campos relacionados en hojas de otros archivos de Excel.

Ante todo esto INCOAYAM como empresa se ha visto en la necesidad de migrar todos sus datos, procedimientos que mantienen en Microsoft Excel a una aplicación personalizada que esté orientada a la web y realice todos los controles que Microsoft Excel no los puede realizar.

### **1.3 PHP**

#### **1.3.1 Historia.**

PHP es una de las muchas tecnologías que se pueden utilizar para afrontar un desarrollo web. PHP surgió de casualidad, como otros muchos grandes descubrimientos a lo largo de la historia, cuando Rasmus Lerdorf buscaba una solución “casera” para el problema de sobrecarga que tenía el servidor web donde alojaba su página personal y currículum. Su intención no era darle un uso más allá del personal y privado. Así creó PHP en el año 1994(Paz).

A partir de ahí, otras personas se interesaron por el pequeño programa y en 1995 Rasmus publicó el código fuente y creó una lista de correos para poder intercambiar ideas, mejoras, correcciones... Desde entonces el lenguaje ha crecido y se ha desarrollado como proyecto open-source. Lo que entonces nació como una lista de correos hoy es una gran comunidad a nivel mundial.

En 1996, solo un año después de su lanzamiento al público, ya se utilizaba en 15.000 sitios web. En 1998 esta presencia en la web ascendía a casi 60.000 dominios,

que suponían alrededor del 1% del total en aquel entonces. Actualmente PHP está presente, aproximadamente, en el 24% de las páginas existentes, una cifra sin duda a tener en cuenta(Rana Negra, 2012).

No sólo de cantidad se trata, sino de calidad e importancia. Muchos de los sitios que conocemos y usamos cada día o con cierta asiduidad están desarrollados en PHP. ¿Quién no ha consultado alguna vez (o muchas) la Wikipedia, ha creado su blog en WordPress o no vive un poquito colgado de Facebook? Todas ellas usan de una u otra forma PHP. Incluso en Twitter, desarrollada en Ruby, se plantearon una migración a PHP por diversas razones, como estabilidad, extensibilidad y escalabilidad de plataforma, incluso por costes en servidores. Migración que finalmente no se llevó a cabo.

A la vista de estos datos, queda claro que la evolución de PHP, acrónimo que ahora responde a Hypertext Preprocessor, ha sido, sin duda, exponencial en todos los sentidos(Paz).

### **1.3.2 ¿Qué es?**

Es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas(González, 2006).

### **1.3.3 ¿Para qué sirve?**

PHP se utiliza para generar páginas web dinámicas. Recordar que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que llamamos páginas dinámicas a aquellas cuyo contenido no es el mismo siempre. Por ejemplo, los contenidos pueden cambiar en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios (González, 2006).

### 1.3.4 ¿Cómo trabaja PHP?

El lenguaje PHP se procesa en servidores, que son potentes ordenadores con un software y hardware especial. Cuando se escribe una dirección tipo `http://www.aprenderaprogramar.com/index.php` en un navegador web como Internet Explorer, Firefox o Chrome (González, 2006).

### 1.3.5 ¿Qué ocurre?

Se envían los datos de la solicitud al servidor que los procesa, reúne los datos (por eso decimos que es un proceso dinámico) y el servidor lo que devuelve es una página HTML como si fuera estática.

El esquema es: Petición de página web al servidor, luego de esto el servidor recibe la petición, reúne la información necesaria consultando a bases de datos o a otras páginas webs, otros servidores. El servidor responde enviando una página web “normal” (estática) pero cuya creación ha sido dinámica, realizando procesos de modo que la página web devuelta no siempre es igual (Lunarwar, 2014).



Figura 1. Tutorial básico PHP

### 1.3.6 Sintaxis.

El intérprete de PHP solo ejecuta el código que se encuentra entre sus delimitadores. Los delimitadores más comunes son `<?php` para abrir una sección PHP y `?>` para cerrarla. El propósito de estos delimitadores es separar el código PHP del resto de código, como por ejemplo el HTML.

Las variables se prefijan con el símbolo del dólar (`$`) y no es necesario indicar su tipo. Las variables, a diferencia de las funciones, distinguen entre mayúsculas y minúsculas. Las cadenas de caracteres pueden ser encapsuladas tanto en dobles comillas como en comillas simples, aunque en el caso de las primeras, se pueden insertar variables en la cadena directamente, sin necesidad de concatenación (Jefry, 2015).

Los comentarios se pueden escribir bien con dos barras al principio de la línea, o con una almohadilla. También permite comentarios multi-línea encapsulados en `/* */`.

En cuanto a las palabras clave, PHP comparte con la mayoría de otros lenguajes con sintaxis C las condiciones con `if`, los bucles con `for` y `while` y los retornos de funciones. Otra característica general de los scripts en PHP es la forma de separar las distintas instrucciones. Para hacerlo, hay que acabar cada instrucción con un punto y coma `;`.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <title> Ejemplo básico PHP</title>
  </head>
  <body>
    <?php
      echo 'Hola mundo';
    ?>
  </body>
</html>
```

**Figura 2. Ejemplo básico PHP**

### 1.3.7 Ventajas.

En el mundo en que vivimos es cada día mayor la presencia de Internet en nuestro entorno y en nuestra vida diaria. Esto lleva consigo un crecimiento en el número de proyectos web que se llevan a cabo, y hace de esta área un campo que merece la atención de emprendedores y nuevos programadores.

Crear un proyecto web puede ser un pequeño paso para un gran proyecto, y elegir la tecnología adecuada puede ser la clave del éxito o fracaso de una idea.

- **Lenguaje totalmente libre y abierto:** Por lo tanto no hay costes de licencias, ni de mantenimientos, ni de ayudas. En principio, su coste se reduciría a instalación, configuración y formación.
- **Curva de aprendizaje muy baja:** Su sintaxis es simple y cumple estándares básicos de la programación orientada a objetos.
- **Los entornos de desarrollo son de rápida y fácil configuración:** No son necesarios complejos entornos de desarrollo, que incluso necesitan su propio periodo de aprendizaje.
- **Fácil despliegue:** Existen paquetes totalmente autoinstalables que integran PHP. Fácilmente se puede programar sin más ayuda que el bloc de notas, todos los IDEs disponibles son gratuitos y los entornos de desarrollo son de rápida y fácil configuración.
- **Fácil acceso a bases de datos:** Ofrece interfaces para el acceso a la mayoría de las bases de datos existentes. Podrás encontrar bases de datos de código abierto, como MySQL, comerciales propietarias como Oracle y además tiene librerías para acceso a datos por ODBC, lo que nos permite comunicar con todas las bases de datos posibles en sistemas Microsoft, como Access o SQL Server. Gracias a los juegos de funciones existentes para cada sistema gestor de base de datos, podremos realizar cualquier acción con los datos que necesitemos para el desarrollo de la más variada gama de aplicación web

- **Comunidad muy grande:** PHP es el lenguaje de programación web más demandado del mercado, es por esto que se ha convertido en la comunidad más grande. Es fácil encontrar una variedad de programas, documentos, código, ejemplos, practicas, es decir posee una amplia gama de soporte a nivel mundial.

### 1.3.8 Desventajas.

Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado. La ofuscación es una técnica que puede dificultar la lectura del código pero no necesariamente impide que el código sea examinado.

Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel, sin embargo este inconveniente se puede minimizar con técnicas de caché tanto en archivos como en memoria.

Las variables al no ser tipificadas dificulta a los diferentes IDEs para ofrecer asistencias para el tipificado del código, aunque esto no es realmente un inconveniente del lenguaje en sí. Esto es solventado por Zend Studio añadiendo un comentario con el tipo a la declaración de la variable(Madecyta, 2012).

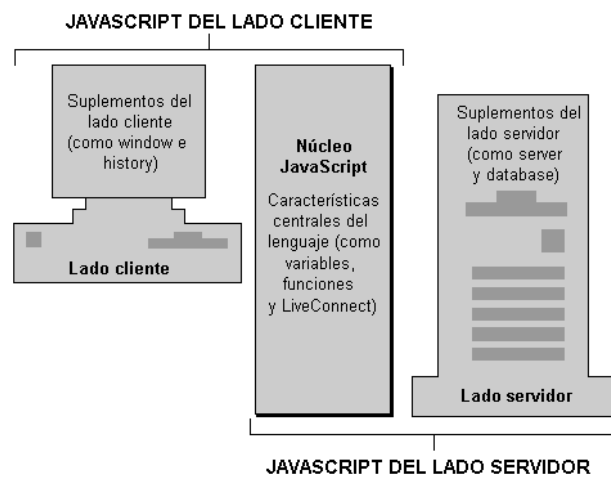
## 1.4 JavaScript (abreviado comúnmente "JS")

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web(Guerra, 2014).

Para entender que es JavaScript tenemos que considerar lo siguiente. Un usuario escribe una dirección web en su navegador o browser, como por ejemplo

<http://www.mipagina.com.ec>. El servidor recibe la petición y como respuesta a esa petición envía al ordenador del usuario código HTML junto a código JavaScript. El código HTML se encarga de que en la pantalla se muestre algo, por ejemplo una imagen, un menú, etc.

El código JavaScript se puede encargar de crear efectos dinámicos en respuesta a acciones del usuario, por ejemplo que se despliegue un menú tipo acordeón cuando el usuario pasa el ratón por encima de un elemento del menú.



**Figura 3. Funcionamiento de javascript**

#### 1.4.1 Historia.

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios complejos.

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje LiveScript.

Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época.

La primera versión de JavaScript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1. Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3. JScript era una copia de JavaScript al que le cambiaron el nombre para evitar problemas legales.

#### **1.4.2 Ventajas.**

JavaScript es una excelente solución para poner en práctica la validación de datos de un formulario en el lado del cliente. Si un usuario omite escribir su nombre en un formulario, una función de validación en JavaScript puede desplegar en pantalla un mensaje popup para hacerle saber al usuario acerca de la omisión. Este tipo de funcionalidades son más ventajosas que tener una rutina de validación del lado del servidor para controlar el error, dado que el servidor en éste caso no tiene que hacer ningún tipo procesamiento de información adicional. Una rutina de ASP o PHP podría ser escrita para lograr la misma tarea pero un formulario desarrollado en JavaScript no permitiría que la información se enviase a menos que se complete correctamente el formulario.

Una de las áreas en la que sobresale radicalmente JavaScript es en la creación de efectos dinámicos tales como imágenes dinámicas y presentaciones de diapositivas, donde su uso se ha convertido algo común hoy en día. Debido a que JavaScript se ejecuta dentro del navegador de los clientes, se puede utilizar para cambiar el aspecto



de la pantalla en el dispositivo de los usuarios después que la página ha sido enviada por el servidor. Esto le permite al desarrollador web crear efectos dinámicos muy impresionantes mejorando así la experiencia que recibe un usuario momento de entrar a un sitio web(Sanchez, 2015).

Al estar alojado en el ordenador del usuario los efectos son muy rápidos y dinámicos. Al ser un lenguaje de programación permite toda la potencia de la programación como uso de variables, condicionales, bucles, etc.

### **1.4.3 Desventajas.**

La seguridad sigue siendo el talón de aquiles de Javascript. Los fragmentos de código de JavaScript una vez añadidos a las páginas web en los servidores, estos son descargados y ejecutados en el navegador del cliente permitiendo así que cierto código malicioso pueda ser ejecutado en la máquina del cliente con el objetivo de explotar alguna vulnerabilidad de seguridad conocida en una de las aplicaciones, navegadores o el mismo sistema operativo. Es verdad que hoy día existen estándares de seguridad que restringen la ejecución de código por parte de los navegadores, pero aun así, se puede ejecutar código que dañe, robe o destruya información del lado del cliente.

Si el usuario tiene desactivado JavaScript en su navegador, no se mostrarán los efectos. No obstante, hoy día la mayoría de los usuarios navegan por la web con JavaScript activado.

Otra desventaja de JavaScript es que este tiende a introducir una cantidad enorme de fragmentos de código en nuestros sitios web. Por suerte, el problema de grandes fragmentos de código JavaScript se resuelve fácilmente mediante el almacenamiento del código JavaScript dentro de archivos separados del código HTML con la extensión. \*.Js, dejando una página web mucho más limpia y legible de cara al desarrollador.

Debido a la tendencia de JavaScript de acrecentar el código de las páginas web, se hace necesario organizar el código JavaScript en archivos separados al código HTML para que los motores de búsqueda (Google) puedan descifrar fácilmente la calidad del contenido de la página web y esta pueda ser indexada correctamente en los resultados de las búsquedas.

Cuando un motor de búsqueda llega a un sitio web buscando contenido de calidad junto con las palabras claves para determinar qué tan útil es la página, lo último que quiere encontrar un motor de búsqueda son cientos de líneas de código JavaScript. Una vez más, este problema se resuelve fácilmente almacenando el código JavaScript en archivos de script con la extensión \*.Js.

#### **1.4.4 Sintaxis y Semántica.**

La sintaxis de JavaScript es muy similar a la de otros lenguajes de programación como Java y C. Las normas básicas que definen la sintaxis de JavaScript son las siguientes:

- **No se tienen en cuenta los espacios en blanco y las nuevas líneas:** como sucede con XHTML, el intérprete de JavaScript ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para entenderlo mejor (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)
- **Se distinguen las mayúsculas y minúsculas:** al igual que sucede con la sintaxis de las etiquetas y elementos XHTML. Sin embargo, si en una página XHTML se utilizan indistintamente mayúsculas y minúsculas, la página se visualiza correctamente, siendo el único problema la no validación de la página. En cambio, si en JavaScript se intercambian mayúsculas y minúsculas el script no funciona.

- **No se define el tipo de las variables:** al crear una variable, no es necesario indicar el tipo de dato que almacenará. De esta forma, una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.
- **No es necesario terminar cada sentencia con el carácter de punto y coma (;):** en la mayoría de lenguajes de programación, es obligatorio terminar cada sentencia con el carácter (;). Aunque JavaScript no obliga a hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma (;).
- **Se pueden incluir comentarios:** los comentarios se utilizan para añadir información en el código fuente del programa. Aunque el contenido de los comentarios no se visualiza por pantalla, sí que se envía al navegador del usuario junto con el resto del script, por lo que es necesario extremar las precauciones sobre la información incluida en los comentarios.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Ejemplo sencillo</title>

<h1 id="header">Esto es JavaScript</h1>

<script>
document.body.appendChild(document.createTextNode('Hola Mundo!'));

var h1 = document.getElementById('header'); // contiene la referencia al tag <h1>
h1 = document.getElementsByTagName('h1')[0]; // accediendo al mismo elemento <h1>
</script>

<noscript>Tu navegador o no admite JavaScript, o ha JavaScript está deshabilitado. </noscript>

</head>
<body>/*...*/</body>
</html>

```

Figura 4. Código Javascript

## 1.5 Netbeans.

NetBeans es un entorno de desarrollo muy completo y profesional. Contiene muchas funcionalidades, para distintos tipos de aplicaciones y para facilitar al máximo la programación, la prueba y la depuración de las aplicaciones que se desarrollan. También incorpora un editor propio.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos(Canedo, 2015).

### **1.5.1 Historia.**

NetBeans comenzó como un proyecto estudiantil en Republica Checa (originalmente llamado Xelfi), en 1996 bajo la tutoría de la Facultad de Matemáticas y Física en la Universidad de Charles en Praga. La meta era escribir un entorno integrado de desarrollo (IDE) para Java parecida a la de Delphi. Xelfi fue el primer entorno integrado de desarrollo escrito en Java, con su primer pre-release en 1997.

Xelfi fue un proyecto divertido para trabajar, ya que las IDEs escritas en Java eran un territorio desconocido en esa época. El proyecto atrajo suficiente interés, por lo que los estudiantes, después de graduarse, decidieron que lo podían convertir en un proyecto comercial. Prestando espacios web de amigos y familiares, formaron una compañía alrededor de esto. Casi todos ellos siguen trabajando en NetBeans(Canedo, 2015).

Tiempo después, ellos fueron contactados por Roman Stanek, un empresario que ya había estado relacionado con varias iniciativas en la Republica Checa. Él estaba buscando una buena idea en que invertir, y encontró en Xelfi una buena oportunidad. Ellos se reunieron, y el negocio surgió.

### **1.5.2 PHP + NetBeans.**

Permite crear aplicaciones Web con PHP 5, un potente debugger integrado y además viene con soporte para Symfony un gran framework MVC escrito en php. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones LAMP o WAMP, están utilizando NetBeans como IDE(Canedo, 2015).

### **1.5.3 Características.**

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente(Canedo, 2015).

Sun Studio, Sun Java Studio Enterprise, y Sun Java Studio Creator de Sun Microsystems han sido todos basados en el IDE NetBeans.

Desde julio de 2006, NetBeans IDE es licenciado bajo la Common Development and Distribution License (CDDL), una licencia basada en la Mozilla Public License(MPL). En octubre de 2007, Sun anunció que NetBeans desde entonces se ofrecerá bajo licenciamiento dual de Licencia CDDL y la GPL versión 2.

## **1.6 HTML HyperText Markup Language.**

HTML es un lenguaje abstracto que aplicaciones pueden usar para representar documentos (se les llama documentos a instancias completas, como lo son las páginas web), y que puede ser transmitido fácilmente por algún medio, como lo es Internet. Los navegadores de Internet procesan e interpretan documentos descritos en HTML usando un analizador de HTML(Castro, 2012).

### **1.6.1 Historia.**

Este lenguaje fue desarrollado por la Organización Europea de Investigación Nuclear (CERN) en el año 1945 con la finalidad de desarrollar un sistema de almacenamiento donde las cosas no se perdieran, que pudieran ser conectadas a través de hipervínculos.

Posteriormente, Douglas Engelbart, diseñó un entorno de trabajo por computadora que poseía un catálogo para facilitar la tarea de búsqueda dentro de un mismo organismo(Castro, 2012).

El tipo de codificación que se utilizó para el desarrollo de este sistema de hipervínculos debía ser comprendido, tanto por ordenadores tontos como por mega-estaciones, por eso fue necesario crear uno absolutamente simples, tanto en lo que respectaba al lenguaje de intercambio (HTML), como el que hacía referencia al protocolo de red (HTTP).

### **1.6.2 Estructura.**

Las páginas HTML se dividen en dos partes: la cabecera y el cuerpo. La cabecera incluye información sobre la propia página, como por ejemplo su título y su idioma. El cuerpo de la página incluye todos sus contenidos, como párrafos de texto e imágenes.

- **<html>**: indica el comienzo y el final de un documento HTML. Ninguna etiqueta o contenido puede colocarse antes o después de la etiqueta `<html>`. En el interior de la etiqueta `<html>` se definen la cabecera y el cuerpo del documento HTML y todo lo que se coloque fuera de la etiqueta `<html>` se ignora.
- **<head>**: delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta `<title>`, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte superior izquierda de la ventana del navegador.
- **<body>**: delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el `<body>` de un documento contiene cientos de etiquetas HTML, mientras que el `<head>` no contiene más que unas pocas.

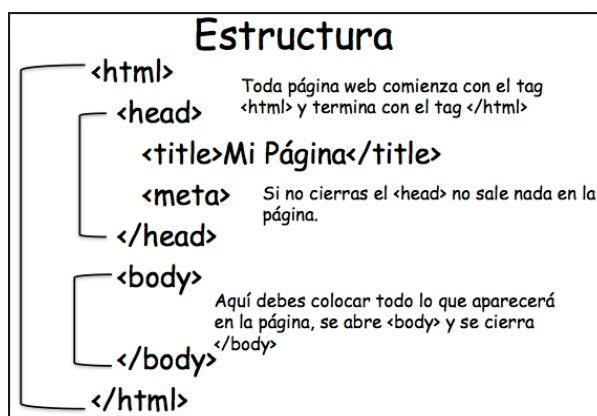


Figura 5. Estructura HTML

### 1.6.3 Ventajas.

- **Facilidad de uso:**Prácticamente cualquier usuario que conozca un poco de computación lo puede aprender fácilmente sin necesidad de tener conceptos básicos de programación.
- **Atractivos:** Siempre los contenidos HTML son más atractivos que los contenidos de un correo en texto plano. Usted puede decidir qué color tendrán los enlaces, el texto o el fondo. También puede incluir imágenes que generarán mayor impacto visual en sus lectores.
- **Mejor respuesta:** Se ha probado que el formato HTML supera en índices de apertura, en muchos casos, al formato de texto plano. La gente está simplemente más interesada en ver un HTML que en leer un simple correo en texto.
- **Diferentes Formatos:** Con archivos de texto plano, sólo se puede escribir de arriba hacia abajo. Mediante el HTML se puede diseñar en 4 o 5 columnas, como en el periódico, e incluir gráficos o imágenes en cualquier parte.

### 1.6.4 Hojas de estilo (CSS).

El lenguaje HTML definía la estructura y la semántica de los documentos y se centraba, sobre todo, en el contenido, ya que surgió como medio de comunicación entre los científicos. Sin embargo, el éxito de la Word Wide Web amplió el abanico de usuarios y poco a poco fue cobrando importancia no sólo el contenido, sino también el diseño y la presentación de la información. Los diseñadores de webs y los editores de páginas HTML comenzaron a integrar nuevas funciones y las sucesivas versiones de HTML incluyeron nuevas etiquetas capaces de realizar esas funciones(Lamarca, 2012).

Sin embargo, la diversidad de plataformas y navegadores que recogían de distinta forma estas nuevas versiones, condujeron a que en la última versión de la



especificación HTML, la 4.0, se volviera hacia el objetivo inicial que era dar importancia a las características que afectaban a la estructura y a la semántica de los documentos, y no a su diseño o aspecto físico, desaconsejando el uso de ciertos elementos y atributos relacionados con la presentación del documento. En esta especificación también se propone delegar todas estas características de presentación y diseño en las hojas de estilo.

Una hoja de estilo (style sheets) no es más que una colección de reglas que afectan a la apariencia de un documento. Estas reglas se refieren al modo en que aparecerá un documento en pantalla cuando el usuario utilice un navegador o explorador gráfico, controlando por ejemplo el color, el fondo, tipo de fuente, apariencia de los bordes, márgenes, alineación y espacio entre caracteres.

El HTML 4.0 desplaza muchas de las funciones realizadas por las anteriores versiones hacia las hojas de estilo, reservando el lenguaje HTML para cuestiones menos formales, relativas a la propia organización del contenido, y lo mismo ocurre con el lenguaje XML(Lamarca, 2012).

Las ventajas que ofrece la utilización de hojas de estilo es que se diseñan de forma independiente al documento HTML y que se pueden aplicar, enteras o alguno de sus niveles, a los documentos que se desee, facilitando la consistencia y homogeneidad en el diseño y la imagen del sitio web. Además, las hojas de estilo se pueden adecuar a los distintos medios de presentación de documentos (pantallas del ordenador, impresoras, televisores, navegadores basados en voz, documentos en Braille...). El usuario también puede elegir qué tipo de diseño prefiere visualizar en un mismo documento: desde un formato más simple o tradicional, a uno más sofisticado. Una ventaja adicional es que reducen el tiempo de carga de las páginas web en el ordenador, puesto que las hojas de estilo se cargan en el navegador o visualizador, de manera que al acceder a un documento que incluya la misma hoja de estilo de otro que se ha "bajado" anteriormente no será necesario volver a cargarla, porque ya se encuentra en el ordenador cliente(Lamarca, 2012).

Actualmente el tipo más común de hoja de estilo es la hoja de estilo en cascada (CSS) o Cascadind Style Sheets.

## 1.7 AJAX Asynchronous JavaScript y XML

Es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones (Wikipedia, 2015).

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML (Wikipedia, 2015).

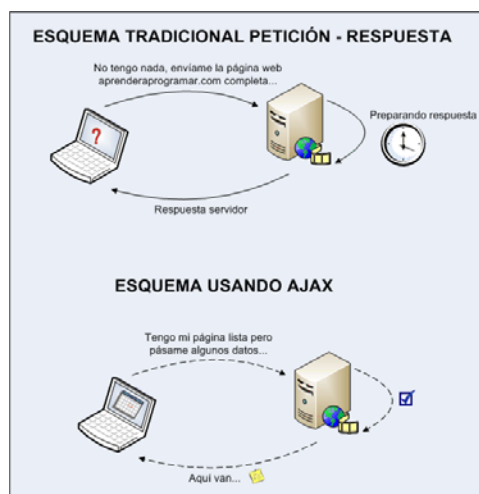


Figura 6. Esquema AJAX

### 1.7.1 Historia.

A pesar de que el término “Ajax” fuese creado en 2005, la historia de las tecnologías que permiten Ajax se remonta a una década antes con la iniciativa de Microsoft en el desarrollo de Scripting Remoto. Sin embargo, las técnicas para la carga asíncrona de contenidos en una página existente sin requerir recarga completa remontan al tiempo del elemento iframe (introducido en Internet Explorer 3 en 1996) y el tipo de elemento layer (introducido en Netscape 4 en 1997, abandonado durante las primeras etapas de desarrollo de Mozilla). Ambos tipos de elemento tenían el atributo src que podía tomar cualquier dirección URL externa, y cargando una página que contenga javascript que manipule la página paterna, pueden lograrse efectos parecidos al Ajax(Wikipedia, 2015).

### **1.7.2 Ventajas.**

- Rapidez en las operaciones.
- Menos carga del servidor (menos transferencia de datos cliente/servidor).
- Menos ancho de banda.
- Soportada por la mayoría de navegadores.
- Interactividad (El usuario no tiene que esperar hasta que lleguen los datos del servidor).
- Portabilidad
- Usabilidad
- Velocidad (Debido a que no hay que recargar la página nuevamente)

### **1.7.3 Desventajas.**

- Se pierde el concepto de “volver a la página anterior”.

- Problemas con navegadores antiguos.
- No funciona si el usuario tiene desactivado el Javascript en su navegador.
- Se requieren conocimiento sobre las tecnologías que forman AJAX.
- Problemas SEO, los buscadores no indexan la información recibida vía AJAX

## 1.8 MySQL

MySQL es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de consulta estructurado.

MySQL se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, MySQL se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (Rouse, 2015).

MySQL, que fue concebido originalmente por la compañía sueca MySQL AB, fue adquirida por Oracle en 2008. Los desarrolladores todavía pueden usar MySQL bajo la Licencia Pública General de GNU (GPL), pero las empresas deben obtener una licencia comercial de Oracle (Rouse, 2015).

### 1.8.1 Lenguajes de programación.

Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi, Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, entre otros; y cada

uno de estos utiliza una interfaz de programación de aplicaciones específica. También existe una interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP(Recargate, 2014).

### **1.8.2 Versiones.**

La evolución de MySQL se produce con las sugerencias de los usuarios, canalizadas por la empresa MySQL AB, que contrata a programadores de todo el mundo para que, a través de Internet, vayan ampliando y mejorando el producto. Las versiones, como es costumbre en este tipo de software libre, se van colgando en Internet para que los usuarios puedan disponer de ellas(Recargate, 2014).

Sin embargo, también como es habitual, hay que distinguir entre versiones de prueba o beta y versiones estables o de producción. Las versiones de prueba o beta están a disposición de los usuarios para ser testadas en busca de posibles mejoras. Por su parte, las versiones estables o de producción son las que ya están preparadas para ser utilizadas a pleno rendimiento.

### **1.8.3 Especificaciones.**

Por ser una base de datos con licencia GPL y que se maneja en la mayoría de aplicaciones web, MySQL trabaja sobre varias plataformas. Entre los diferentes sistemas operativos están: AIX, BSD, FreeBSD, HP-UX, Kurisu OS, GNU/Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, QNX, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8 y Windows Server.

### **1.8.4 Licencia.**

MySQL provee licenciamiento dual, se puede obtener el código fuente bajo la licencia GPL v2 o se puede comprar una licencia comercial.

He aquí algunas cosas que están permitidas bajo la licencia GPL:

- La licencia GPL le permite manejar un negocio con fines de lucro usando MySQL.
- La licencia GPL permite modificar el código fuente de MySQL en la forma que desee.
- La licencia GPL le permite vender y distribuir MySQL.
- La licencia GPL le permite redistribuir las modificaciones de MySQL.

Todo esto que es permitido por la licencia GPL viene con algunas restricciones, pero esas restricciones (en general) son sólo para evitar que el software siga siendo “libre”. Así, por ejemplo, si vendemos o redistribuimos, hay que hacerlo bajo la licencia GPL también. No se puede despojar a la parte GPL o gravar el software y luego pasar a una versión menos libre del software para los demás.

Aquí hay algunas cosas que la licencia GPL no requiere:

- La licencia GPL no le obliga a redistribuir las modificaciones de MySQL.
- La licencia GPL no requiere de una licencia GPL del software que simplemente se conecta a MySQL.
- La licencia GPL no requiere que todo el software en su empresa/organización también sea GPL.

Con todo esto mencionado se puede concluir con algunos citados:

- Es necesaria una licencia comercial si desea modificar MySQL y redistribuir el resultado como software no libre.

- Es necesaria una licencia comercial si desea incorporar MySQL dentro de un programa no-Libre. Tenga en cuenta que incorporar no es lo mismo que “hacer una conexión”.

Ya que estas cosas no están permitidas por la licencia GPL, lo que necesita es comprar el derecho de usar el código fuente de MySQL bajo una licencia no-GPL. Ahí es donde las licencias duales entran en juego.

## **1.9 Apache.**

Apache es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas operativos y mantener su excelente rendimiento(Garcia, 2013).

Desde el año 1996, es el servidor web más popular del mundo, debido a su estabilidad y seguridad. Apache sigue siendo desarrollado por la comunidad de usuarios desarrolladores que trabaja bajo la tutela de Apache Software Foundation.

Apache es utilizado principalmente, para realizar servicio a páginas web, ya sean estáticas o dinámicas. Este estupendo servidor se integra a la perfección con otras aplicaciones, creando el famoso paquete XAMP con Perl, Python, MySQL y PHP, junto a cualquier sistema operativo, que por lo general es Linux, Windows o Mac OS(Garcia, 2013).

Entre las principales características de Apache, se encuentran las siguientes:

- Soporte de seguridad SSL y TLS.
- Puede realizar autenticación de datos utilizando SGDB.
- Puede dar soporte a diferentes lenguajes, como Perl, PHP, Python y tcl.

### 1.9.1 Historia.

En febrero del año 1995, el servidor web más popular era un servidor desarrollado por el NCSA (National Center for Supercomputing Applications de la Universidad de Illinois). Sin embargo en el año 1994, al dejar la NCSA el principal desarrollador del servidor Rob McCool, la evolución del programa había quedado seriamente comprometida. La responsabilidad del desarrollo recayó en los responsables de sitios web, que introdujeron mejoras progresivas en sus servidores. Un grupo de ellos, utilizando el correo electrónico como herramienta principal de coordinación, se pusieron de acuerdo para poner en común estas mejoras en forma de parches, dos de ellos, Cliff Skolnick y Brian Behlendorf, iniciaron una lista de correo, un espacio para compartir información y un servidor en California donde los desarrolladores más importantes pudiesen trabajar. A principios del año siguiente, 8 programadores fundaron lo que había de ser el Grupo Apache(Garcia, 2013).

Éstos, utilizando como base de trabajo el servidor NCSA 1.3, incorporaron las correcciones de errores publicadas y las mejoras más importantes que encontraron y probaron el resultado final en sus servidores. Después publicaron lo que había de ser la 1a versión oficial del servidor Apache (la 0.6.2, en abril del año 1995). Casualmente, por esas fechas, la NCSA reemprendió el desarrollo de su servidor NCSA.

En aquel momento el desarrollo de Apache continuó por 2 líneas paralelas. Por un lado, algunos desarrolladores siguieron trabajando en el Apache 0.6.2 para llegar a la serie 0.7, incorporando diversas mejoras. Otro grupo reescribió por completo el código de la primera versión, creando una nueva arquitectura de tipo modular. En julio del año 1995 migraron a esta nueva arquitectura las mejoras desarrolladas para Apache 0.7, haciéndose público como Apache 0.8.

El día 1 de diciembre del año 1995, apareció Apache 1.0, que incluía abundante documentación y muchas mejoras en forma de módulos que se podían incrustar. Después, Apache sobrepasó al servidor NCSA como el más popular en Internet,



posición que ha mantenido hasta hoy. En el año 1999 los miembros del Grupo Apache fundaron la Apache Software Foundation, que da soporte de tipo legal y financiero al desarrollo del servidor Apache y los proyectos relacionados que ha ido surgiendo(García, 2013).

### 1.9.2 Ventajas.

- **Apertura:** Apache Web Server es un software de código abierto. Esto significa que la programación que impulsa el software puede ser consultada y editada por cualquiera en el mundo. Este diseño abierto permite a cualquier programador crear una solución personalizada basada en el programa núcleo de Apache, o ampliar las funciones del software. Muchas de estas extensiones personalizadas se han escrito y están disponibles de forma gratuita. Si la función de medida perfecta no existe todavía, se puede crear fácilmente(Cordero, 2014).
- Al ser de código abierto, Apache se actualiza constantemente. Muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web. El enfoque de código abierto significa que los usuarios no tienen que esperar a que versiones completas del software sean liberadas. En su lugar, pequeñas actualizaciones pueden ser instaladas tan pronto como estén disponibles(Cordero, 2014).
- **Costo:** Una segunda ventaja relacionada con el diseño de código abierto de Apache es su costo. El servidor web Apache es completamente gratuito y puede ser descargado por cualquier persona en el mundo. Por el contrario, la competencia, como la tecnología de servidor web de Windows Server 2008 puede tener un costo mínimo de US\$470, con un precio de más de 1,000 dólares para las instalaciones más avanzadas(Cordero, 2014).

- **Funcionalidades:** A pesar de su costo mínimo, Apache Web Server tiene un gran conjunto de funcionalidades de gran alcance. Estas características principales, junto con las extensiones creadas por programadores de todo el mundo, ayudan a que la plataforma Apache sea competitiva incluso frente a rivales de alto precio. Apache ha incorporado en su soporte a una amplia gama de lenguajes de programación web, como Perl, PHP y Python. Estos lenguajes son fáciles de aprender y se pueden utilizar para crear potentes aplicaciones en línea. Apache también incluye soporte "SSL" y "TLS". Estos son los protocolos para enviar datos encriptados a través de Internet, y son importantes en el desarrollo de tiendas seguras en línea y otras aplicaciones que requieren privacidad(Cordero, 2014).
- **Soporte:** Apache Web Server cuenta con una gran comunidad de usuarios de soporte. A diferencia de muchas compañías de software que se encargan de todo el soporte a los programas desde un solo lugar, el soporte técnico de Apache se extiende a lo largo de múltiples localizaciones, empresas, y foros. Este modelo de distribución del soporte permite a los usuarios obtener respuestas a preguntas técnicas casi las 24 horas al día, no importa dónde se encuentren(Cordero, 2014).
- Al ser de código abierto, Apache está conectado a muchos usuarios que son capaces de crear parches y correcciones de errores técnicos muy rápidamente. Tan pronto como se encuentra un problema, los usuarios de todo el mundo comunican y aportan soluciones. El resultado de este soporte de la comunidad es software que es muy estable y bien mantenido(Cordero, 2014).
- **Portabilidad:** Apache Web Server es muy portable. Esto significa que se puede instalar en una amplia variedad de servidores y sistemas operativos. Apache es capaz de ejecutarse en todas las versiones del sistema operativo UNIX, así como con Linux, los sistemas operativos Windows NT y MacOS. Desde un punto de vista de hardware, Apache puede ser utilizado en cualquier servidor con procesador de la serie Intel 80x86. Si Apache está

siendo utilizado en un sistema operativo Unix o Linux, casi cualquier tipo de procesador es compatible. En general, Apache es uno de los sistemas de servidores más adaptables disponibles en la actualidad(Cordero, 2014).

## CAPÍTULO II

### 2 PLANEACION

Para la elaboración del Plan de Entrega, es necesario en primer lugar identificar las Historias de Usuario, las mismas que especifican los requerimientos del Software y representan la unidad funcional en un proyecto XP, en donde el cliente describe las características que el sistema debe poseer.

#### 2.1 Historias de Usuario.

Las Historias de Usuario tienen el mismo propósito que los casos de uso y constituyen una técnica utilizada en el desarrollo de proyectos XP, las mismas que permiten especificar los requerimientos de usuario necesarios para el desarrollo de la aplicación, mediante la utilización de tarjetas en las cuales se describe las características que el sistema debe poseer. La identificación de las historias de usuario depende de la complejidad del sistema y son descompuestas en tareas de programación (task card) que son asignadas al programador para ser implementadas durante una iteración. Es conveniente especificar al menos una historia por cada característica importante.

Para la presentación de las Historias de Usuario, se ha considerado la siguiente plantilla.

HISTORIA DE USUARIO			
NUMERO:		USUARIO:	
NOMBRE:			
MODIFICACION DE HISTORIA N°:		ITERACCION ASIGNADA:	
PRIORIDAD EN NEGOCIO:		PUNTOS ESTIMADOS:	
RIESGO EN DESARROLLO:		PUNTOS REALES:	
DESARROLADOR ENCARGADO:			
DESCRIPCION:			
OBSERVACION:			

**Figura 7. Plantilla de historias de usuario**

### 2.1.1 Módulos del Sistema:

A continuación se presenta las Historias de usuario para el presente proyecto, las mismas que al ser priorizadas se agrupan en los siguientes módulos:

- ***Módulo N° 1 Administración del Sistema***
  - Registro de información de usuarios.
  - Registro de prioridades de los proyectos.
  - Registro de los perfiles de usuario.
  - Registro de las líneas de negocio.
  - Registro de perfiles del trabajador.
  - Registro de estados del proyecto.
  - Registro de contactos.
  
- ***Módulo N° 2 Mantenimiento de Kits***
  - Ingreso de la cabecera del kit.
  - Ingreso de materia prima al kit.
  - Ingreso de mano de obra al kit.

- ***Módulo N° 3 Gestión control de Ofertas***
  - Crear una oferta.
  - Ingreso de estados a la oferta.
  - Ingreso de contactos a la oferta.
  
- ***Módulo N° 4 Gestión de cotizaciones***
  - Ingreso de datos a la cabecera de la cotización.
  - Añadir kits a la cotización.
  - Añadir otros gastos a la cotización.
  - Añadir datos informativos a la cotización.
  
- ***Módulo N° 5 Planificación del proyecto***
  - Ingreso de datos al proyecto.
  - Ingreso de clientes al proyecto.
  - Ingreso de otros datos al proyecto.
  - Ingreso de tareas al proyecto.

### **2.1.2 Especificación de historias de usuario por módulos del sistema**

#### **Modulo N°1 Administración del sistema**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU01	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro de usuarios		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	0,50
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	0
<b>DESARROLADOR ENCARGADO:</b>	Andrés Peralta		
<b>DESCRIPCION:</b>	Se realiza el control de todos los usuarios que van a interactuar con el sistema con sus distintos perfiles, datos personales, fecha de ingreso, fecha de caducidad.		
<b>OBSERVACION:</b>	Se realizara todas las operaciones CRUD(Create, , Update, Delete), para la eliminar el usuario se hará un borrado logico el usuario estará atado a un usuario del ERP.		

**Figura 8. Historia de usuario registro de usuarios**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU02	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro de prioridades de los proyectos		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Baja	<b>PUNTOS ESTIMADOS:</b>	0,25
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	0
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el control de los distintos tipos de estado de prioridad (nombre, nombre corto, descripcion, estado) que tendran todos los proyectos.		
<b>OBSERVACION:</b>	Para el eliminado se utilizan borrados lógicos		

**Figura 9. Historia de usuario prioridades de proyectos**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU03	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro de perfiles de usuario		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	0,50
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	0
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el mantenimiento de los perfiles(nombre, nombre corto, descripcion, permiso, estado), los cuales tendrán los permisos a los distintos modulos del sistema.		
<b>OBSERVACION:</b>	Para el eliminado se utilizan borrados lógicos		

**Figura 10. Historia de usuario perfil de usuarios**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU04	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro de líneas de negocio		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Baja	<b>PUNTOS ESTIMADOS:</b>	0,25
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el mantenimiento de las distintas líneas de negocio (nombre, nombre corto, descripcion, estado), que la empresa tiene en sus proyectos.		
<b>OBSERVACION:</b>	Para el eliminado se utilizan borrados lógicos		

**Figura 11. Historia de usuario líneas de negocio**



<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU05	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro perfil del trabajador		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Baja	<b>PUNTOS ESTIMADOS:</b>	0,50
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el mantenimiento de los tipos de trabajadores con sus perfiles(tipo de trabajador, nombre del perfil, costo por dia, estado), que la empresa tiene en sus proyectos.		
<b>OBSERVACION:</b>	Para el eliminado se utilizan borrados lógicos		

**Figura 12. Historia de usuario perfil de trabajador**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU06	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro de estados de proyectos		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Baja	<b>PUNTOS ESTIMADOS:</b>	0,25
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el mantenimiento de los tipos de estado(nombre corto, nombre, descripcion, permisos, estado) que tendra el proyecto.		
<b>OBSERVACION:</b>	Los permisos estan dados para controlar la aparicion o no aparicion del proyecto en los distintos modulos del sistema.		

**Figura 13. Historia de usuario estados de proyectos**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU07	<b>USUARIO:</b>	GERENTE
<b>NOMBRE:</b>	Registro de contactos		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Primera
<b>PRIORIDAD EN NEGOCIO:</b>	Baja	<b>PUNTOS ESTIMADOS:</b>	0,50
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el mantenimiento de contactos(nombre, telefono 1, telefono 2, celular 1, celular 2, e-mail, direccion, estado) que interactuan en los proyectos.		
<b>OBSERVACION:</b>	Para el eliminado se utilizan borrados lógicos		

**Figura 14. Historia de usuario contactos**

## Modulo N°2 Mantenimiento de Kits

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU10	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Ingreso cabecera kit		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Segunda
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	0,50
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andrés Peralta		
<b>DESCRIPCION:</b>	Se gestiona el ingreso de la cabecera del kit(código, nombre, unidad, línea de negocio, rendimiento, costo indirecto, estado, total del kit)		
<b>OBSERVACION:</b>	Rendimiento = días/kits Costo indirecto = dato variable que viene dado cada mes.		

**Figura 15. Historia de usuario ingreso cabecera kit**

HISTORIA DE USUARIO			
NUMERO:	HU11	USUARIO:	EMPLEADOS
NOMBRE:	Ingreso de materia prima en el kit		
MODIFICACION DE HISTORIA N°:	0	ITERACCION ASIGNADA:	Segunda
PRIORIDAD EN NEGOCIO:	Alta	PUNTOS ESTIMADOS:	0,75
RIESGO EN DESARROLLO:	Media	PUNTOS REALES:	
DESARROLADOR ENCARGADO:	Andres Peralta		
DESCRIPCION:	Se realiza el ingreso de la materia prima que tiene el kit (codigo, nombre, unidad, cantidad, costo, total)		
OBSERVACION:	Los datos de materia prima seran cojidos de la base de datos de productos del ERP		

Figura 16. Historia de usuario ingreso materia prima

HISTORIA DE USUARIO			
NUMERO:	HU12	USUARIO:	EMPLEADOS
NOMBRE:	Registro de mano de obra en el kit		
MODIFICACION DE HISTORIA N°:	0	ITERACCION ASIGNADA:	Segunda
PRIORIDAD EN NEGOCIO:	Alta	PUNTOS ESTIMADOS:	0,75
RIESGO EN DESARROLLO:	Media	PUNTOS REALES:	
DESARROLADOR ENCARGADO:	Andres Peralta		
DESCRIPCION:	Se realiza el ingreso de la mano de obra que tiene el kit (codigo, nombre, perfil, costo perfil, cantidad, obreros dia, costo mano de obra)		
OBSERVACION:	$\text{obrerros dia} = (\text{dias/kit}) * \text{cantidad}$ $\text{costo mano de obra} = \text{obrerros dia} * \text{costo perfil}$		

Figura 17. Historia de usuario registro de mano de obra

Modulo N°3 Gestión control de ofertas.

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU20	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Crear una oferta		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Tercera
<b>PRIORIDAD EN NEGOCIO:</b>	Media	<b>PUNTOS ESTIMADOS:</b>	1,00
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andrés Peralta		
<b>DESCRIPCION:</b>	Se gestiona el mantenimiento del control de ofertas(código, nombre de la oferta, línea de negocio, descripción).		
<b>OBSERVACION:</b>	La Oferta tiene al menos un estado inicial.		

**Figura 18. Historia de usuario crear una oferta**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU21	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Ingreso de estados a la oferta		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Tercera
<b>PRIORIDAD EN NEGOCIO:</b>	Media	<b>PUNTOS ESTIMADOS:</b>	0,50
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el ingreso del detalle de estados(estado, fecha estado, observacion, siguiente estado, fecha siguiente estado, siguiente gestion).		
<b>OBSERVACION:</b>	Una oferta tiene uno o mas estados De esta manera se controla los estados de los posibles proyectos que reciban cotizaciones.		

**Figura 19. Historia de usuario ingreso de estados a la oferta**

HISTORIA DE USUARIO			
NUMERO:	HU22	USUARIO:	EMPLEADOS
NOMBRE:	Ingreso de contactos a la oferta		
MODIFICACION DE HISTORIA N°:	0	ITERACCION ASIGNADA:	Tercera
PRIORIDAD EN NEGOCIO:	Media	PUNTOS ESTIMADOS:	0,50
RIESGO EN DESARROLLO:	Baja	PUNTOS REALES:	
DESARROLADOR ENCARGADO:	Andres Peralta		
DESCRIPCION:	Se realiza el ingreso de contactos de la base de datos de contactos, si no hubiera se creara directamente uno nuevo		
OBSERVACION:	Una oferta tiene uno o varios contactos.		

Figura 20. Historia de usuario ingreso de contactos a la oferta

#### Modulo N°4 Gestión cotizaciones.

HISTORIA DE USUARIO			
NUMERO:	HU30	USUARIO:	EMPLEADOS
NOMBRE:	Ingreso datos a la cabecera de la cotización		
MODIFICACION DE HISTORIA N°:	0	ITERACCION ASIGNADA:	Cuarta
PRIORIDAD EN NEGOCIO:	Alta	PUNTOS ESTIMADOS:	1,00
RIESGO EN DESARROLLO:	Media	PUNTOS REALES:	
DESARROLADOR ENCARGADO:	Andrés Peralta		
DESCRIPCION:	Se ingresa los datos principales de la cotización(código, oferta, nombre cotización, descripción, tiempo de ejecución, fecha, días de vigencia, estado)		
OBSERVACION:	Las ofertas aparecerán de acuerdo al estado de las mismas. Existen tres estados para la cotización emitida, aceptada, rechazada, por default estará en estado emitida.		

Figura 21. Historia de usuario cabecera de la cotización

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU31	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Añadir kits a la cotización		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Cuarta
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	2,00
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se ingresan uno mas kits(cantidad)que estan ya ingresados en la base de datos, ademas se brinda la posibilidad de cambiar los datos de costos en materia prima y mano de obra.		
<b>OBSERVACION:</b>	Cuando se haga una modificacion en los costos de algun kit solo afectara para dicha cotizacion mas no para el kit en si.		

**Figura 22. Historia de usuario kits a la cotización**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU32	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Añadir otros gastos a la cotización		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Cuarta
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	2,00
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	La cotizacion maneja tres tipos de gastos adicionales: Aumenta: Aumenta una cantidad, unidad, descripcion, valor a la cotizacion Recalcula: Quita un porcentaje al valor total de la cotizacion. Extras: Aumenta el valor sumado con la ganacia de un producto de la base de datos.		
<b>OBSERVACION:</b>	Aumenta: El valor de aumento se reparte a los diferentes detalles de la cotizacion y no aparece como un detalle mas. Recalcula: El porcentaje quitado crea un detalle adicional en la cotizacion. Extras: Aumenta un detalle en la cotizacion.		

**Figura 23. Historia de usuario gastos a la cotización**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU33	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Añadir datos informativos en la cotizacion		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Cuarta
<b>PRIORIDAD EN NEGOCIO:</b>	Baja	<b>PUNTOS ESTIMADOS:</b>	0,75
<b>RIESGO EN DESARROLLO:</b>	Baja	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	La cotizacion maneja datos adicionales(incluye, no incluye, forma de pago, otros)que son ingresados como parte de la misma.		
<b>OBSERVACION:</b>	Los campos seran suficientemente grandes como para poder ingresar saltos de pagina.		

**Figura 24. Historia de usuario datos informativos a la cotización**

### Modulo N°5 Planificación del proyecto.

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU40	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Ingreso de datos al proyecto		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Quinta
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	0,75
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andrés Peralta		
<b>DESCRIPCION:</b>	Se gestionara los datos del proyecto(contrato, cotización, numero, nombre, dirección, fecha, teléfonos obra, forma de pago, estado).		
<b>OBSERVACION:</b>	Al momento de seleccionar la Cotización se lo hace de acuerdo al estado de la misma.		

**Figura 25. Historia de usuario datos al proyecto**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU41	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Ingreso de personas involucradas al proyecto		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Quinta
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	0,75
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se gestionara las personas involucradas en el Proyecto(tipo, nombre, telefono1, telefono2, celular1, celular2, dirección, observación).		
<b>OBSERVACION:</b>	Las personas involucradas son de uno o mas y los tipos seran: Residente de obra, encargado de la negociacion, gerente del proyecto o fiscalizador.		

**Figura 26. Historia de usuario personas involucradas al proyecto**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU42	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Ingreso de clientes al proyecto		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Quinta
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	1,50
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se agrega la informacion acerca de los clientes(tipo cliente, nombre, ruc, direccion, telefono, observacion).		
<b>OBSERVACION:</b>	Los clientes involucrados pueden ser de uno o mas y los tipos son : contratante o facturar.		

**Figura 27. Historia de usuario clientes al proyecto**



<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU43	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Registro de otros datos al proyecto		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Quinta
<b>PRIORIDAD EN NEGOCIO:</b>	Media	<b>PUNTOS ESTIMADOS:</b>	1,50
<b>RIESGO EN DESARROLLO:</b>	Media	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se brinda la posibilidad de agregar otros tipos de personas secundarias al proyecto(datos personales).		
<b>OBSERVACION:</b>	Las personas secundarias involucradas al proyecto pueden ser de cero o mas y son de dos tipos: pagos o recepcion de facturas.		

**Figura 28. Historia de usuario otros datos al proyecto**

<b>HISTORIA DE USUARIO</b>			
<b>NUMERO:</b>	HU44	<b>USUARIO:</b>	EMPLEADOS
<b>NOMBRE:</b>	Ingreso de tareas al proyecto		
<b>MODIFICACION DE HISTORIA N°:</b>	0	<b>ITERACCION ASIGNADA:</b>	Quinta
<b>PRIORIDAD EN NEGOCIO:</b>	Alta	<b>PUNTOS ESTIMADOS:</b>	1,50
<b>RIESGO EN DESARROLLO:</b>	Alta	<b>PUNTOS REALES:</b>	
<b>DESARROLADOR ENCARGADO:</b>	Andres Peralta		
<b>DESCRIPCION:</b>	Se realiza el registro de tareas(nombre, fecha inicio, fecha fin, tarea antescesora, observaciones, kits, responsable) de acuerdo a la informacion que contiene la cotizacion.		
<b>OBSERVACION:</b>	Los kits vienen dados por la cotizacion y no pueden sobrepasar el numero que se estimo en un principio de la cotizacion. El responsable puede ser de dos tipos: Interno: Es dado por los empleados registrados en el ERP. Externo: Agente externo a la empresa.		

**Figura 29. Historia de usuario tareas al proyecto**

### 2.1.3 Valoración de historias de usuario

Como punto importante en el desarrollo del proyecto en especial de la Planificación de la Entrega del proyecto, se considera la realización de la valoración de todas las Historias de Usuario.

Esto se hace especificando un tiempo estimado para la elaboración de cada una, en base a una semana de 5 días y un día de 5 horas, dicho reporte se lo representa a continuación con la siguiente tabla.

**Tabla 1**  
*Estimación de tiempo de historias de usuarios*

MODULO	NRO.	HISTORIA DE USUARIO	TIEMPO ESTIMADO		
			SEMANAS ESTIMADAS	DIAS ESTIMADAS	HORAS ESTIMADAS
Administración del Sistema	HU01	Registro de usuarios	0,50	2,50	12,50
	HU02	Registro de prioridades de los proyectos	0,25	1,25	6,25
	HU03	Registro de perfiles de usuario	0,50	2,50	12,50
	HU04	Registro de líneas de negocio	0,25	1,25	6,25
	HU05	Registro perfil del trabajador	0,50	2,50	12,50
	HU06	Registro de estados de proyectos	0,25	1,25	6,25
	HU07	Registro de contactos	0,50	2,50	12,50
Mantenimiento de Kits	HU10	Ingreso cabecera kit	0,50	2,50	12,50
	HU11	Ingreso de materia prima en el kit	0,75	3,75	18,75
	HU12	Registro de mano de obra en el kit	0,75	3,75	18,75
Gestión de Control de Ofertas	HU20	Crear una oferta	1,00	5,00	25,00
	HU21	Ingreso de estados a la oferta	0,50	2,50	12,50
	HU22	Ingreso de contactos a la oferta	0,50	2,50	12,50
Gestión de Cotizaciones	HU30	Ingreso datos a la cabecera de la cotización	1,00	5,00	25,00
	HU31	Añadir kits a la cotización	2,00	10,00	50,00
	HU32	Añadir otros gastos a la cotización	2,00	10,00	50,00
	HU33	Añadir datos informativos en la cotización	0,75	3,75	18,75
Planificación del Proyecto	HU40	Ingreso de datos al proyecto	0,75	3,75	18,75
	HU41	Ingreso de personas involucradas al proyecto	0,75	3,75	18,75
	HU42	Ingreso de clientes al proyecto	1,50	7,50	37,50
	HU43	Registro de otros datos al proyecto	1,50	7,50	37,50
	HU44	Ingreso de tareas al proyecto	1,50	7,50	37,50
<b>TIEMPO TOTAL ESTIMADO :</b>			<b>18,50</b>	<b>92,50</b>	<b>462,50</b>

## 2.2 Plan de Entrega

Para la elaboración del Plan de Entrega del presente proyecto y aplicando los parámetros de desarrollo bajo la metodología XP, se establece el tiempo calendario de acuerdo a un mes de 4 semanas, una semana de 5 días y un día de 5 horas.

### 2.2.1 Tiempo Calendario

Para la estimación del Tiempo Calendario, se presenta la siguiente plantilla como modelo a utilizar en horas, días y semanas, la misma que permitirá calcular el esfuerzo utilizado para de desarrollo del presente proyecto.

**Tabla 2.**  
*Tabla tiempo calendario*

<b>Horas Calendario</b>	<b>Días calendario</b>	<b>Semanas Calendario</b>
5 horas (horas diarias que se va a dedicar al desarrollo del proyecto)	5 días (días laborables que se va a dedicar al desarrollo del proyecto)	4 semanas (semanas al mes que se va a dedicar al desarrollo del proyecto)

### 2.2.2 Esfuerzo de Desarrollo

Se considera: Personas en el equipo: 1 persona; por lo tanto:

Esfuerzo por semana de desarrollo: 1personas = 1semanas

Esfuerzo por días de desarrollo: 1personas = 5días

Esfuerzo por horas calendario de desarrollo: 1personas = 5horas

En consecuencia a este análisis, se tiene:

**Tabla 3.**  
*Esfuerzo de desarrollo en base a una persona*

<b>Personas en el Equipo</b>	<b>Horas de Esfuerzo de Desarrollo</b>	<b>Días de Esfuerzo de Desarrollo</b>	<b>Semanas de Esfuerzo de Desarrollo</b>
1 persona	5 horas	5 días	1 semana

### **2.2.3 Elaboración del Plan de Entrega**

Determinado el tiempo calendario y estimado el esfuerzo de desarrollo, se procede a la elaboración del plan de entrega, mediante la utilización de la estimación por historia de usuario.

## **2.3 Cuadro de entregables**

Los cuadros de entregables, constituyen un acuerdo entre el cliente, el cual establece la prioridad de cada historia de usuario, de acuerdo con el valor que aporta para el negocio y el desarrollador que es quien estima el esfuerzo requerido para la implementación de cada historias de usuario, así como el tiempo empleado para la entrega de cada iteración.

### **2.3.1 Entregables en XP**

Los entregables según la metodología que estamos empleando, son lanzamientos cortos y frecuentes correspondientes a las tareas necesarias para completar la implementación de cada una de las iteraciones.

Para el presente proyecto se especifican como cuadros de entregables los siguientes historiales:

- **Historial de versiones por historia de usuario.**
- **Historial de seguimiento de CRC.**
- **Historial de Seguimiento de la Ejecución de Pruebas de Aceptación.**
- **Historial de Seguimiento de Iteraciones.**

- **Historial de Seguimiento de Entrega Final.**

**Tabla 4.**  
*Historial de versiones por historia de usuario*

ITERACION	NR. O.	HISTORIA DE USUARIO	PRIORIDAD (ENTREGA)	ACTIVIDAD (Nueva / Corrección / Mejora)	DEPENDENCIA (Nro. Historia de Usuario)	RIESGO (Alto / Medio / Bajo)	VERSION	ESTADO DE DESARROLLO	PRUEBAS
Primera	HU01	Registro de usuarios	1	Nueva	NA	Baja	1	Completo	Aprobado
	HU02	Registro de prioridades de los proyectos	1	Nueva	NA	Baja	1	Completo	Aprobado
	HU03	Registro de perfiles de usuario	1	Nueva	HU01	Baja	1	Completo	Aprobado
	HU04	Registro de líneas de negocio	1	Nueva	NA	Baja	1	Completo	Aprobado
	HU05	Registro perfil del trabajador	1	Nueva	NA	Baja	1	Completo	Aprobado
	HU06	Registro de estados de proyectos	1	Nueva	NA	Baja	1	Completo	Aprobado
	HU07	Registro de contactos	1	Nueva	NA	Baja	1	Completo	Aprobado
Segunda	HU10	Ingreso cabecera kit	2	Nueva	HU04	Media	1	Completo	Aprobado
	HU11	Ingreso de materia prima en el kit	2	Nueva	NA	Media	1	Completo	Aprobado
	HU12	Registro de mano de obra en el kit	2	Nueva	NA	Media	1	Completo	Aprobado
Tercera	HU20	Crear una oferta	3	Nueva	HU04	Baja	1	Completo	Aprobado
	HU21	Ingreso de estados a la oferta	3	Nueva	NA	Baja	1	Completo	Aprobado
	HU22	Ingreso de contactos a la oferta	3	Nueva	HU07	Baja	1	Completo	Aprobado
Cuarta	HU30	Ingreso datos a la cabecera de la cotización	4	Nueva	NA	Media	1	Completo	Aprobado
	HU31	Añadir kits a la cotización	4	Nueva	HU10	Media	1	Completo	Aprobado
	HU32	Añadir otros gastos a la cotización	4	Nueva	NA	Media	1	Completo	Aprobado
	HU33	Añadir datos informativos en la cotización	4	Nueva	NA	Baja	1	Completo	Aprobado
Quinta	HU40	Ingreso de datos al proyecto	5	Nueva	HU07,HU30	Media	1	Completo	Aprobado
	HU41	Ingreso de personas involucradas al proyecto	5	Nueva	HU07	Media	1	Completo	Aprobado
	HU42	Ingreso de clientes al proyecto	5	Nueva	NA	Media	1	Completo	Aprobado
	HU43	Registro de otros datos al proyecto	5	Nueva	HU07	Media	1	Completo	Aprobado
	HU44	Ingreso de tareas al proyecto	5	Nueva	HU31	Alta	1	Completo	Aprobado

**Tabla 5.**  
*Historial de seguimiento de CRC*

NRO.	ESCENARIO	TARJETA CRC	TERMINADO (SI/NO)	NIVEL DE AVANCE	OBSERVACION	ESTADO
CRC01	Registro de usuarios	Usuarios	SI	100%	Se realiza las operaciones Crud de los usuarios	APROBADO
CRC02	Registro de prioridades de los proyectos	Prioridades de Proyecto	SI	100%	Se realiza las operaciones Crud de las prioridades	APROBADO
CRC03	Registro de perfiles de usuario	Perfiles de Usuario	SI	100%	Se realiza las operaciones Crud de los perfiles	APROBADO
CRC04	Registro de líneas de negocio	Líneas de Negocio	SI	100%	Se realiza las operaciones Crud de las líneas	APROBADO
CRC05	Registro perfil del trabajador	Perfiles de Trabajador	SI	100%	Se realiza las operaciones Crud de los trabajadores	APROBADO
CRC06	Registro de estados de proyectos	Estados de Proyecto	SI	100%	Se realiza las operaciones Crud de los estados	APROBADO
CRC07	Registro de contactos	Contactos	SI	100%	Se realiza las operaciones Crud de los contactos	APROBADO
CRC10	Ingreso cabecera kit	Cabeceras de Kit	SI	100%	Se realiza las operaciones Crud de la cabecera del kit	APROBADO
CRC11	Ingreso de materia prima en el kit	Materias Primas en Kit	SI	100%	Se realiza las operaciones Crud de la materia prima	APROBADO
<b>CONTINUA</b>						→

CRC12	Registro de mano de obra en el kit	Manos de Obra en Kit	SI	100%	Se realiza las operaciones Crud de la mano de obra	APROBADO
CRC20	Crear una oferta	Ofertas	SI	100%	Se realiza las operaciones Crud de la oferta	APROBADO
CRC21	Ingreso de estados a la oferta	Estados en Oferta	SI	100%	Se realiza las operaciones Crud de los estados de oferta	APROBADO
CRC22	Ingreso de contactos a la oferta	Contactos en Oferta	SI	100%	Se realiza las operaciones Crud de los contactos de oferta	APROBADO
CRC30	Ingreso datos a la cabecera de la cotización	Cabeceras de Cotización	SI	100%	Se realiza las operaciones Crud de la cabecera de la cotización	APROBADO
CRC31	Añadir kits a la cotización	Kits en la Cotización	SI	100%	Se realiza las operaciones Crud de los kits de cotización	APROBADO
CRC32	Añadir otros gastos a la cotización	Otros Gastos en la Cotización	SI	100%	Se realiza las operaciones Crud de los gastos de cotización	APROBADO
CRC33	Añadir datos informativos en la cotización	Datos Informativos en la Cotización	SI	100%	Se realiza las operaciones Crud de los datos informativos	APROBADO
CRC40	Ingreso de datos al proyecto	Cabecera de Proyectos	SI	100%	Se realiza las operaciones Crud de los datos del proyecto	APROBADO
CRC41	Ingreso de personas involucradas al proyecto	Personas Involucradas en Proyecto	SI	100%	Se realiza las operaciones Crud de las personas involucradas	APROBADO
CRC42	Ingreso de clientes al proyecto	Clientes en Proyecto	SI	100%	Se realiza las operaciones Crud de los clientes del proyecto	APROBADO
CRC43	Registro de otros datos al proyecto	Otros Datos en Proyecto	SI	100%	Se realiza las operaciones Crud de otros datos del proyecto	APROBADO

**Tabla 6.**  
**Historial de Seguimiento de la Ejecución de Pruebas de Aceptación**

ESCENARIO	CASO DE PRUEBA	NRO. HISTORIA DE USUARIO	NRO. CASO DE PRUEBA	RESULTADO DE LA PRUEBA
Registro de usuarios	Guardar Usuario	HU01	CP01.01	EXITOSA
	Buscar Usuario	HU01	CP01.02	EXITOSA
	Editar Usuario	HU01	CP01.03	EXITOSA
	Eliminar Usuario	HU01	CP01.04	EXITOSA
	Resetear Clave	HU01	CP01.05	EXITOSA
Registro de prioridades de los proyectos	Guardar Prioridad de Proyecto	HU02	CP02.01	EXITOSA
	Buscar Prioridad de Proyecto	HU02	CP02.02	EXITOSA
	Editar Prioridad de Proyecto	HU02	CP02.03	EXITOSA
	Eliminar Prioridad de Proyecto	HU02	CP02.04	EXITOSA
Registro de perfiles de usuario	Guardar Perfil	HU03	CP03.01	EXITOSA
	Buscar Perfil	HU03	CP03.02	EXITOSA
	Editar Perfil	HU03	CP03.03	EXITOSA
	Eliminar Perfil	HU03	CP03.04	EXITOSA
Registro de líneas de negocio	Guardar Línea	HU04	CP04.01	EXITOSA
	Buscar Línea	HU04	CP04.02	EXITOSA
	Editar Línea	HU04	CP04.03	EXITOSA
	Eliminar Línea	HU04	CP04.04	EXITOSA
Registro perfil del trabajador	Guardar Perfil de Trabajador	HU05	CP05.01	EXITOSA
	Buscar Perfil Trabajador	HU05	CP05.02	EXITOSA
	Editar Perfil Trabajador	HU05	CP05.03	EXITOSA
	Eliminar Perfil de Trabajador	HU05	CP05.04	EXITOSA
Registro de estados de proyectos	Guardar Estado	HU06	CP06.01	EXITOSA
	Buscar Estado	HU06	CP06.02	EXITOSA
	Editar Estado	HU06	CP06.03	EXITOSA
	Eliminar Estado	HU06	CP06.04	EXITOSA
Registro de contactos	Guardar Contacto	HU07	CP07.01	EXITOSA

CONTINUA



	Buscar Contacto	HU07	CP07.02	EXITOSA
	Editar Contacto	HU07	CP07.03	EXITOSA
	Eliminar Contacto	HU07	CP07.04	EXITOSA
Ingreso cabecera kit	Guardar Kit	HU10	CP10.01	EXITOSA
	Búsqueda Kit	HU10	CP10.02	EXITOSA
	Editar Kit	HU10	CP10.03	EXITOSA
	Eliminar Kit	HU10	CP10.04	EXITOSA
Ingreso de materia prima en el kit	Agregar detalle Materia Prima	HU11	CP11.01	EXITOSA
	Editar detalle Materia Prima	HU11	CP11.02	EXITOSA
	Eliminar detalle Materia Prima	HU11	CP11.03	EXITOSA
Registro de mano de obra en el kit	Agregar detalle Mano de Obra	HU12	CP12.01	EXITOSA
	Editar detalle Mano de Obra	HU12	CP12.02	EXITOSA
	Eliminar detalle Mano de Obra	HU12	CP12.03	EXITOSA
Crear una oferta	Guardar Oferta	HU20	CP20.01	EXITOSA
	Buscar Oferta	HU20	CP20.02	EXITOSA
	Editar Oferta	HU20	CP20.03	EXITOSA
Ingreso de estados a la oferta	Agregar Estado a la Oferta	HU21	CP21.01	EXITOSA
	Eliminar Estado de la Oferta	HU21	CP21.02	EXITOSA
Ingreso de contactos a la oferta	Agregar Contacto a la Oferta	HU22	CP22.01	EXITOSA
	Crear Contacto	HU22	CP22.02	EXITOSA
	Eliminar Contacto de Oferta	HU22	CP22.03	EXITOSA
Ingreso datos a la cabecera de la cotización	Guardar Cotización	HU30	CP30.01	EXITOSA
	Buscar Cotización	HU30	CP30.02	EXITOSA
	Editar Cotización	HU30	CP30.03	EXITOSA
	Copiar Cotización	HU30	CP30.04	EXITOSA
	Calcular Cotización	NA	CP30.05	EXITOSA
Añadir kits a la cotización	Agregar Kit	HU31	CP31.01	EXITOSA
	Consultar Kit	HU31	CP31.02	EXITOSA
	Editar Kit	HU31	CP31.03	EXITOSA
	Eliminar Kit	HU31	CP31.04	EXITOSA
	Calcular Kit	NA	CP31.05	EXITOSA
	Editar Materia Prima	HU31	CP31.06	EXITOSA
Añadir otros gastos a la cotización	Editar Mano de Obra	HU31	CP31.07	EXITOSA
	Agregar Gasto Aumenta Cotización	HU32	CP32.01	EXITOSA
	Eliminar Gasto Aumenta Cotización	HU32	CP32.02	EXITOSA
	Agregar Gasto Recalcula Cotización	HU32	CP32.03	EXITOSA
	Eliminar Gasto Recalcula Cotización	HU32	CP32.04	EXITOSA
	Agregar Gasto Extras Cotización	HU32	CP32.05	EXITOSA
Añadir datos informativos en la cotización	Eliminar Gasto Extras Cotización	HU32	CP32.06	EXITOSA
	Agregar Dato Informativo	HU33	CP33.01	EXITOSA
	Eliminar Dato Informativo	HU33	CP33.02	EXITOSA
Ingreso de datos al proyecto	Guardar Proyecto	HU40	CP40.01	EXITOSA
	Buscar Proyecto	HU40	CP40.02	EXITOSA
	Editar Proyecto	HU40	CP40.03	EXITOSA
Ingreso de personas involucradas al proyecto	Agregar Personas Involucradas	HU41	CP41.01	EXITOSA

CONTINUA



	Eliminar Personas Involucradas	HU41	CP41.02	EXITOSA
Ingreso de clientes al proyecto	Agregar Cliente	HU42	CP42.01	EXITOSA
	Eliminar Cliente	HU42	CP42.02	EXITOSA
Registro de otros datos al proyecto	Agregar Otro Dato	HU43	CP43.01	EXITOSA
	Eliminar Otro Dato	HU43	CP43.02	EXITOSA
	Guardar Tarea	HU44	CP44.01	EXITOSA
Ingreso de tareas al proyecto	Editar Tarea	HU44	CP44.02	EXITOSA
	Eliminar Tarea	HU44	CP44.03	EXITOSA
	Agregar Kit a la Tarea	HU44	CP44.04	EXITOSA
	Eliminar Kit de Tarea	HU44	CP44.05	EXITOSA
	Agregar Responsable a la Tarea	HU44	CP44.06	EXITOSA
	Eliminar Responsable de Tarea	HU44	CP44.07	EXITOSA

**Tabla 7.**  
*Historial de seguimiento de iteraciones*

ITERACIÓN	NRO. HISTORIA DE USUARIO	HISTORIA DE USUARIO	FECHA PLANIFICACIÓN	LANZAMIENTO	ESTADO DE DESARROLLO	PRUEBAS
Primera	HU01	Registro de usuarios	3 de noviembre de 2014	5 de noviembre de 2014	20 de noviembre de 2014	Completo
	HU02	Registro de prioridades de los proyectos	5 de noviembre de 2014	6 de noviembre de 2014	20 de noviembre de 2014	Completo
	HU03	Registro de perfiles de usuario	6 de noviembre de 2014	11 de noviembre de 2014	20 de noviembre de 2014	Completo
	HU04	Registro de líneas de negocio	11 de noviembre de 2014	12 de noviembre de 2014	20 de noviembre de 2014	Completo
	HU05	Registro perfil del trabajador	12 de noviembre de 2014	14 de noviembre de 2014	20 de noviembre de 2014	Completo
	HU06	Registro de estados de proyectos	17 de noviembre de 2014	18 de noviembre de 2014	20 de noviembre de 2014	Completo
	HU07	Registro de contactos	18 de noviembre de 2014	20 de noviembre de 2014	20 de noviembre de 2014	Completo
Segunda	HU10	Ingreso cabecera kit	20 de noviembre de 2014	25 de noviembre de 2014	18 de diciembre de 2014	Completo
	HU11	Ingreso de materia prima en el kit	25 de noviembre de 2014	28 de noviembre de 2014	18 de diciembre de 2014	Completo
	HU12	Registro de mano de obra en el kit	1 de diciembre de 2014	4 de diciembre de 2014	18 de diciembre de 2014	Completo
Tercera	HU20	Crear una oferta	4 de diciembre de 2014	11 de diciembre de 2014	18 de diciembre de 2014	Completo
	HU21	Ingreso de estados a la oferta	11 de diciembre de 2014	16 de diciembre de 2014	18 de diciembre de 2014	Completo
	HU22	Ingreso de contactos a la oferta	16 de diciembre de 2014	18 de diciembre de 2014	18 de diciembre de 2014	Completo
Cuarta	HU30	Ingreso datos a la cabecera de la cotización	18 de diciembre de 2014	25 de diciembre de 2014	11 de marzo de 2015	Completo
	HU31	Añadir kits a la cotización	25 de diciembre de 2014	8 de enero de 2015	11 de marzo de 2015	Completo
	HU32	Añadir otros gastos a la cotización	8 de enero de 2015	22 de enero de 2015	11 de marzo de 2015	Completo
	HU33	Añadir datos informativos en la cotización	22 de enero de 2015	28 de enero de 2015	11 de marzo de 2015	Completo
Quinta	HU40	Ingreso de datos al proyecto	22 de enero de 2015	3 de febrero de 2015	11 de marzo de 2015	Completo
	HU41	Ingreso de personas involucradas al proyecto	3 de febrero de 2015	6 de febrero de 2015	11 de marzo de 2015	Completo
	HU42	Ingreso de clientes al proyecto	9 de febrero de 2015	18 de febrero de 2015	11 de marzo de 2015	Completo
	HU43	Registro de otros datos al proyecto	18 de febrero de 2015	27 de febrero de 2015	11 de marzo de 2015	Completo
	HU44	Ingreso de tareas al proyecto	2 de marzo de 2015	11 de marzo de 2015	11 de marzo de 2015	Completo



**Tabla 8.**  
**Historial de seguimiento de entrega final**

ITERACIÓN	NRO. HISTORIA DE USUARIO	HISTORIA DE USUARIO	ITERACIÓN ASIGNADA					ENTREGA ESTIMADA			VERSIÓN	OBSERVACIÓN	ESTADO	
			1	2	3	4	5	1	2	3				
Administración del sistema	HU01	Registro de usuarios	X					X			1	Se entregan sin novedades	Aprobado	
	HU02	Registro de prioridades de los proyectos	X					X			1	Se entregan sin novedades	Aprobado	
	HU03	Registro de perfiles de usuario	X					X			1	Se entregan sin novedades	Aprobado	
	HU04	Registro de líneas de negocio	X					X			1	Se entregan sin novedades	Aprobado	
	HU05	Registro perfil del trabajador	X					X			1	Se entregan sin novedades	Aprobado	
	HU06	Registro de estados de proyectos	X					X			1	Se entregan sin novedades	Aprobado	
	HU07	Registro de contactos	X					X			1	Se entregan sin novedades	Aprobado	
Mantenimiento de kits	HU10	Ingreso cabecera kit		X						X	1	Se entregan sin novedades	Aprobado	
	HU11	Ingreso de materia prima en el kit		X						X	1	Se entregan sin novedades	Aprobado	
	HU12	Registro de mano de obra en el kit		X						X	1	Se entregan sin novedades	Aprobado	
Gestión control de oferta	HU20	Crear una oferta			X					X	1	Se entregan sin novedades	Aprobado	
	HU21	Ingreso de estados a la oferta			X					X	1	Se entregan sin novedades	Aprobado	
	HU22	Ingreso de contactos a la oferta			X					X	1	Se entregan sin novedades	Aprobado	
Gestión de cotizaciones	HU30	Ingreso datos a la cabecera de la cotización				X					X	1	Se entregan sin novedades	Aprobado
	HU31	Añadir kits a la cotización				X					X	1	Se entregan sin novedades	Aprobado
	HU32	Añadir otros gastos a la cotización				X					X	1	Se entregan sin novedades	Aprobado
	HU33	Añadir datos informativos en la cotización				X					X	1	Se entregan sin novedades	Aprobado
	HU40	Ingreso de datos al proyecto					X				X	1	Se entregan sin novedades	Aprobado
Planificación del Proyecto	HU41	Ingreso de personas involucradas al proyecto					X				X	1	Se entregan sin novedades	Aprobado
	HU42	Ingreso de clientes al proyecto					X				X	1	Se entregan sin novedades	Aprobado
	HU43	Registro de otros datos al proyecto					X				X	1	Se entregan sin novedades	Aprobado
	HU44	Ingreso de tareas al proyecto					X				X	1	Se entregan sin novedades	Aprobado

## 2.4 Planificación de iteraciones.

Cada iteración necesita ser planificada, a esto se llama planificación iterativa, para ello se desarrolla en un entorno que involucra lo posible y lo deseable, mediante la utilización de reglas que gobiernan la entrega de cada iteración con la finalidad de darle mayor valor al producto, buscando además reducir el tiempo, la inversión y el riesgo que se presente en el periodo de desarrollo de la aplicación. Además para evitar cuellos de botella y fomentar la propiedad colectiva del código.

Para la planificación, es necesario identificar las iteraciones que han sido previamente establecidas, así como las historias de usuario consideradas primordiales

para la implementación del proyecto; con la finalidad de poder determinar las tareas necesarias para la elaboración de cada historia, mediante la estimación de tiempo necesario para la ejecución de cada una de ellas.

Este patrón permite añadir agilidad al proceso de desarrollo y evita realizar tareas que no estén programadas (programación just-in-time), enfocando la concentración en lo que realmente es necesario y no en incrementar la funcionalidad antes de tiempo.

La planificación en iteraciones y el diseño iterativo permite mantener discusiones continuas acerca de los problemas y avances realizados en las tareas, fomentando así la comunicación entre las partes involucradas y ayudando a la optimización de tiempo y esfuerzo empleado para el correcto desarrollo del proyecto.

#### **2.4.1 Tareas de ingeniería.**

Para proceder a la elaboración de las historias de usuario ya mencionadas en el proyecto, se realiza varias tareas de diseño, las mismas que son sugeridas por la metodología XP para la planificación de iteraciones del proyecto de desarrollo y se muestran a continuación.

- **Diseño de Datos:** Permite mostrar el alcance de la aplicación, brindando una visión clara de las tareas que el proyecto va a realizar mediante métodos y especificaciones que cumplan con el propósito de la aplicación.
- **Diseño tarjetas CRC:** Las tarjetas CRC aportan en la tarea de diseño del proyecto y representan escenarios provenientes de las historias de usuario, los cuales especifican las posibles métodos o funciones que tendrá la aplicación, además de identificar las responsabilidades y colaboradores para finalmente ser traducidas en relaciones en su respectiva iteración.
- **Documentación de Escenario:** Definido el alcance de la aplicación, el siguiente paso es la documentación de escenarios, que consiste en almacenar

los diseños que enfocan la solución del sistema en un repositorio o a su vez reunirlos mediante documentación.

- **Programación:** La programación consiste en el desarrollo del código de la implementación de cada uno de los escenarios (historias de usuario).
- **Ejecución de pruebas:** Las pruebas en XP, están estrechamente relacionadas con la planificación de iteraciones; razón por la cual deben ser elaboradas primero.
- **Documentación de las pruebas de aceptación:** Las iteraciones deben ser elaboradas secuencialmente para mantener un seguimiento adecuado del proceso de desarrollo de la aplicación, razón por la cual las pruebas de aceptación son manejadas mediante documentos que constatan la correcta implementación de cada historia de usuario.

Considerando que la metodología XP brinda la posibilidad de incluir herramientas que apoyen al proyecto en cada una de sus etapas de desarrollo, por tal motivo la planificación de Iteraciones, se documentará mediante la utilización de la herramienta OpenSource denominada GanttProject, el mismo que muestra las actividades desarrolladas para cada historia de usuario como se muestra a continuación.

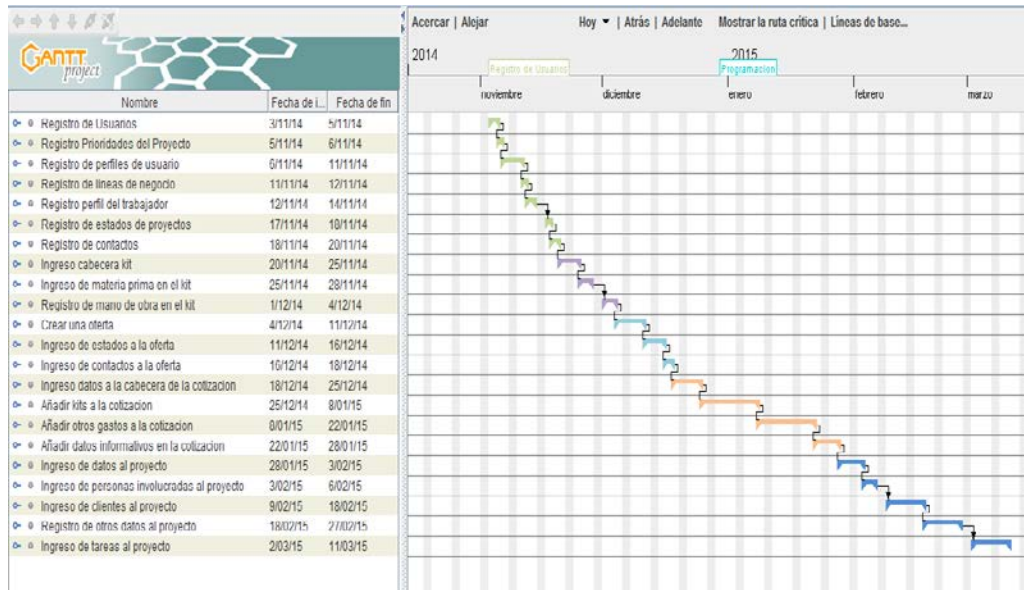


Figura 30. Planificación de iteración por historia de usuario

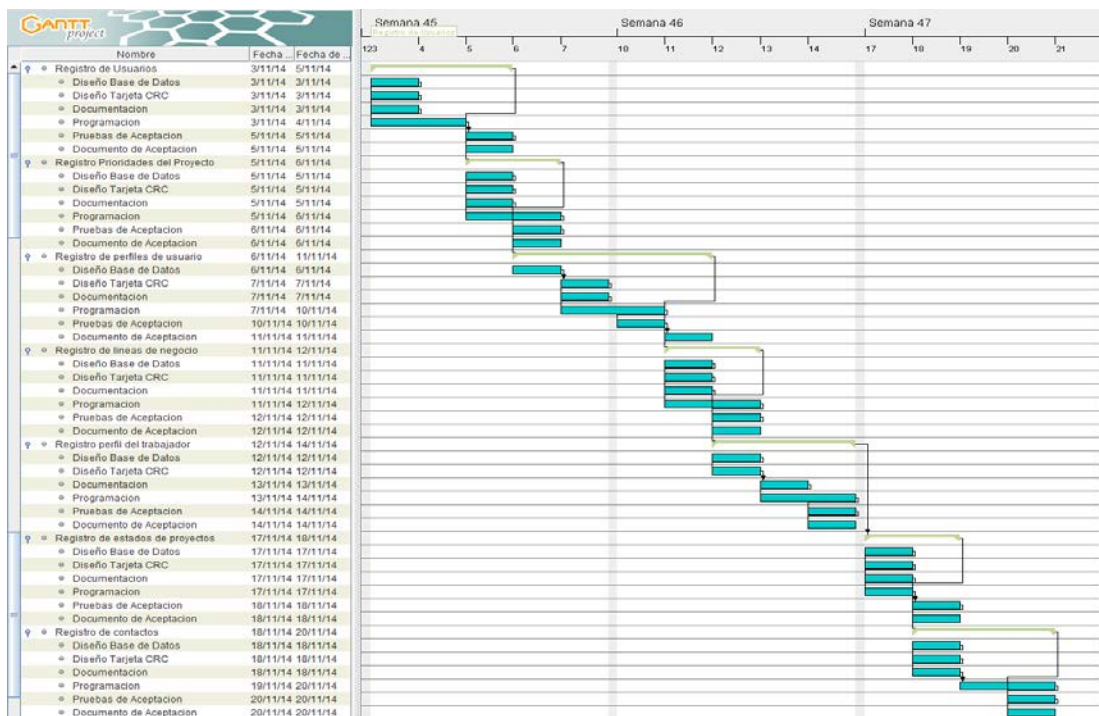


Figura 31. Planificación de la primera iteración

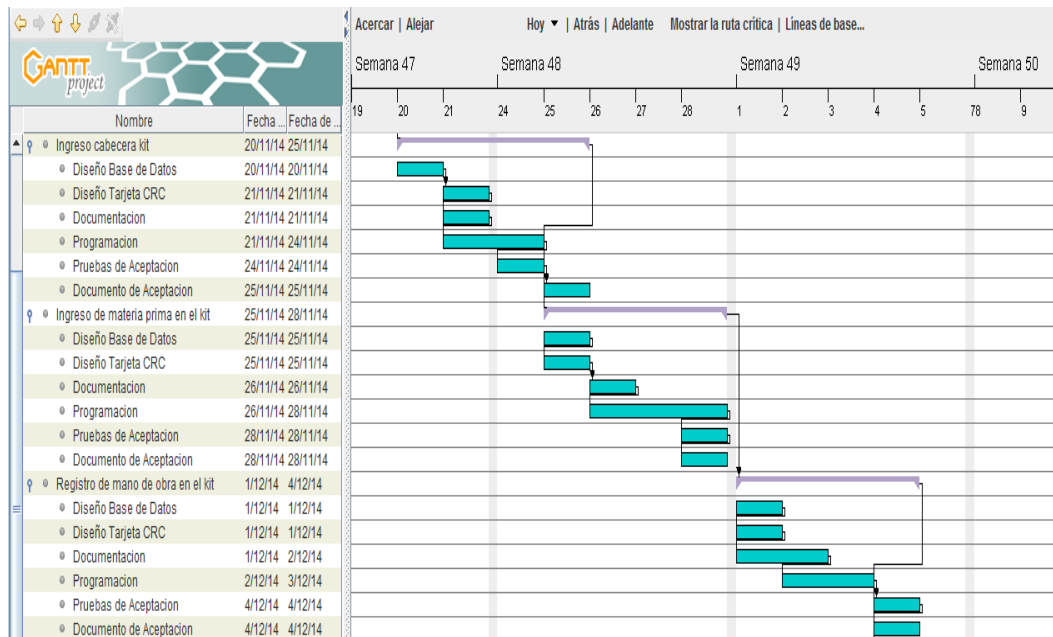


Figura 32. Planificación de la segunda iteración

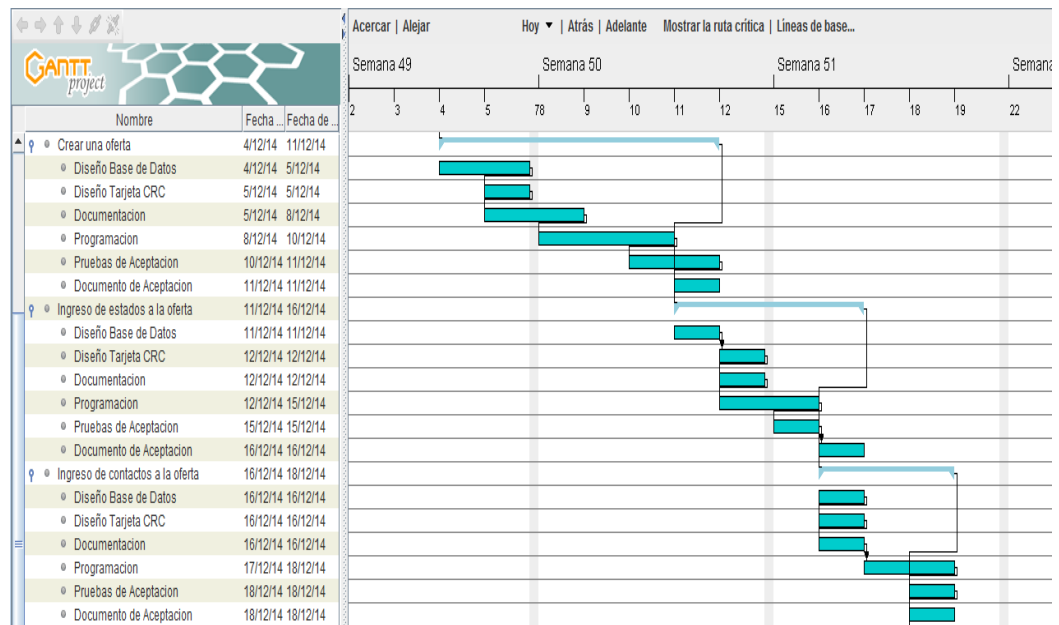


Figura 33. Planificación de la tercera iteración

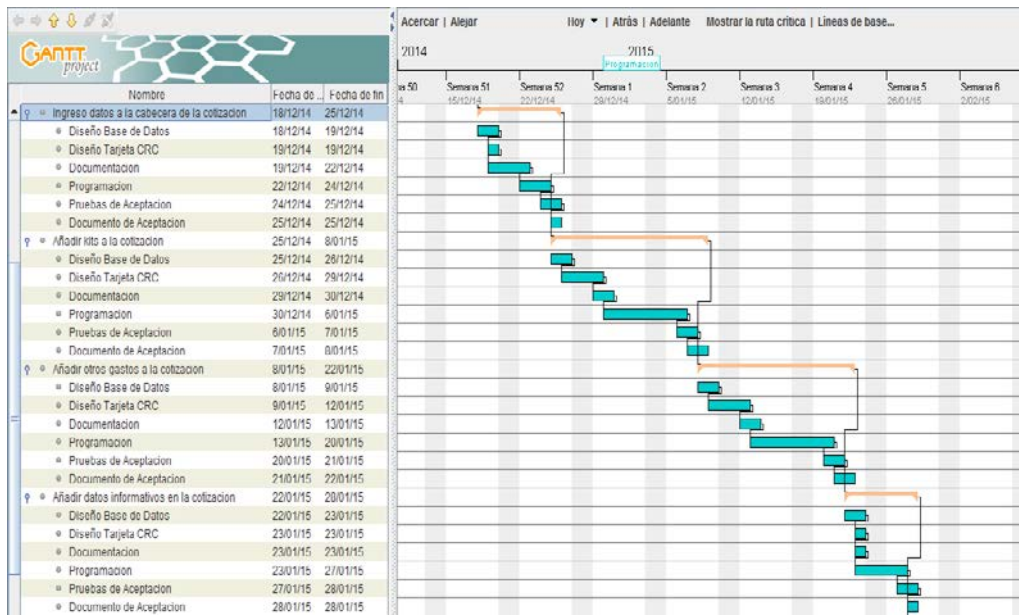


Figura 34. Planificación de la cuarta iteración

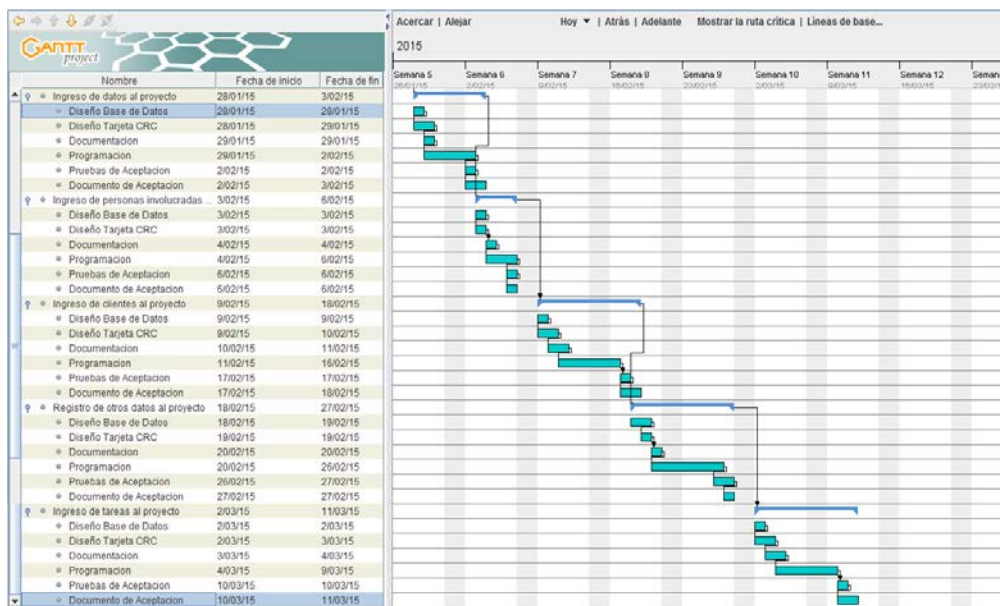


Figura 35. Planificación de la quinta iteración

## CAPÍTULO III

### 3 DISEÑO

#### 3.1 Diseño simple

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. Un diseño simple siempre toma menos tiempo para finalizarlo que un complejo. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Siempre es más rápido y más barato reemplazar ahora el código complejo antes que gastemos más tiempo en él.

Hay que mantener las cosas lo más simples posibles tanto como sea posible a través de no agregar nuevas funcionalidades antes de que sean agendadas.

Por esto se hace el uso de un pequeño framework que maneja la empresa Sisconti, la cual ayuda a generar de manera más eficiente código html para poder presentarlo en la capa de presentación. Además de esto posee las conexiones a las bases de datos tanto con Informix y MySQL.

#### 3.2 Metáforas

En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema. Dicha arquitectura se asume evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora.

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema. La práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.

Para el caso práctico del desarrollo del sistema, la empresa INCOAYAM posee una serie de archivos realizados en Microsoft Excel que explican cómo es el funcionamiento de los diferentes módulos que vamos a crear.

Cada archivo posee datos con fórmulas y sus respectivas explicaciones, además de que los campos de cada celda están bien definidos con sus nombres propios, los cuales ayudaran a identificar variables, funciones, entidades, en el desarrollo del proyecto.

### **3.3 Refactorización**

Para el desarrollo del proyecto, como se mencionó anteriormente el framework que posee la empresa Sisconti tiene algunas funciones útiles que servirán para validaciones, u otras utilidades. Para si fuera el caso poder reutilizar el código y si es necesario modificarlo de acuerdo a la necesidad del proyecto.

Estas funciones se encuentran en librerías dentro del proyecto, estas librerías son:

- común.php
- formulario.php

### **3.4 No agregar funcionalidades antes de lo planeado**

Siempre estamos tentados en agregar funcionalidades ahora antes que después debido a que sabemos exactamente como agregarla o porque haría al sistema mucho mejor. Parecería que fuera más rápido agregarlas ahora pero nosotros debemos recordarnos constantemente que no las necesitamos ahora realmente y quizás nunca las necesitemos.

Para el desarrollo del proyecto se optó por realizar en primer lugar las interfaces con las que va a interactuar el usuario para posterior y con el ok del cliente poder



realizar toda la codificación de cada módulo según sea la iteración que se esté desarrollando.

### **3.4.1 Especificación de la arquitectura del sistema**

La arquitectura que utiliza el framework es de 3 capas por lo que se va a seguir utilizando la misma y que se detalla a continuación.

#### **Capa 1: Componente Arquitectónico Contenedor de Librerías.**

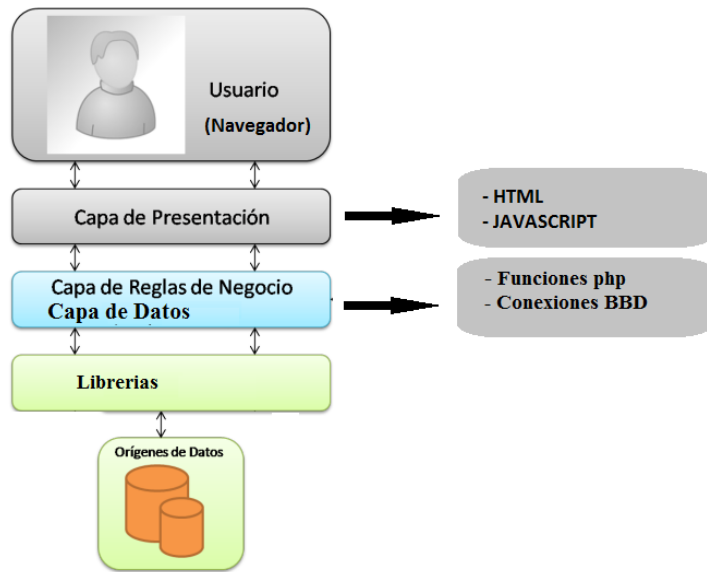
En este componente se gestiona todos los objetos como clases, conexiones, archivos de configuración, librerías, utilidades, la cual se comunica directamente con la capa 2.

#### **Capa 2: Componente Arquitectónico de Acceso a Datos y Negocio.**

En este componente se gestiona todos los objetos de tipo dato con la base de datos del sistema, además de toda la lógica del negocio. Esta capa interactúa con la de contenedor de librerías y la de presentación.

#### **Capa 3: Componente Arquitectónico de Presentación:**

Este componente basado en páginas html contiene la implementación de la interface del sistema.



**Figura 36. Esquema n-capas**

### 3.5 Tarjetas de Responsabilidad Colaboración (CRC)

La mayor ventaja de las tarjetas CRC es permitir reducir el modo de pensar y apreciar la tecnología de objetos. Se definen responsabilidades para cada tarjeta en donde se muestran los problemas que van a ser resueltos, mediante la utilización de una frase que inicia con un verbo activo. Otro aspecto a tomar en cuenta son los colaboradores que expresan dependencias entre objetos; colaboradores son los que envían o reciben mensajes, se les pide información o realizar alguna acción.

Para la documentación de tarjetas CRC, se especifica la utilización de la siguiente plantilla.

TARJETA CRC		
NUMERO:		ESCENARIO:
NOMBRE CRC:		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:
OBSERVACIONES:		

Figura 37. Plantilla base tarjeta CRC

### 3.5.1 Tarjetas CRC primera iteración.

#### Registro de usuarios.

- Realizar operaciones CRUD.

TARJETA CRC		
NUMERO:	CRC01	ESCENARIO:
NOMBRE CRC:	Usuarios	
	Registro de usuarios	
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:
<ul style="list-style-type: none"> <li>- Obtener el tipo de perfil.</li> <li>- Guardar un registro de usuario.</li> <li>- Obtener un registro de usuario.</li> <li>- Actualizar un registro de usuario.</li> <li>- Eliminar un registro de usuario.</li> <li>- Cambiar la clave de usuario.</li> </ul>	Perfiles	<ul style="list-style-type: none"> <li>- ObtenerPerfil</li> <li>- GuardarUsuario</li> <li>- ObtenerUsuario</li> <li>- ActualizarUsuario</li> <li>- EliminarUsuario</li> <li>- CambiarClave</li> </ul>
OBSERVACIONES:		
- Se registran todos los usuarios que van a interactuar con el sistema.		

Figura 38. Tarjeta CRC\_usuarios

#### Registro de prioridades de los proyectos

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC02	ESCENARIO:	Registro de prioridades de los proyectos
NOMBRE CRC:	Prioridades de Proyecto		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un registro de una prioridad de proyecto.</li> <li>- Obtener un registro de una prioridad de proyecto.</li> <li>- Actualizar un registro de una prioridad de proyecto.</li> <li>- Eliminar un registro de una prioridad de proyecto.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarPrioridad</li> <li>- ObtenerPrioridad</li> <li>- ActualizarPrioridad</li> <li>- EliminarPrioridad</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todas las prioridades que puede poseer un proyecto.</li> </ul>			

**Figura 39. Tarjeta CRC\_prioridades\_de\_proyecto**

### Registro de perfiles de usuario

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC03	ESCENARIO:	Registro de perfiles de usuario
NOMBRE CRC:	Perfiles de Usuario		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un registro de un perfil de usuario.</li> <li>- Obtener un registro de un perfil de usuario.</li> <li>- Actualizar un registro de un perfil de usuario.</li> <li>- Eliminar un registro de un perfil de usuario.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarPerfil</li> <li>- ObtenerPerfil</li> <li>- ActualizarPerfil</li> <li>- EliminarPerfil</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todos los perfiles que van a tener los usuario.</li> <li>- Un perfil pueden tener cero o varios usuarios.</li> <li>- Cada perfil tiene los permisos de acceso a los diferentes modulos del sistema.</li> </ul>			

**Figura 40. Tarjeta CRC\_perfiles\_de\_usuario**

### Registro de líneas de negocio

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC04	ESCENARIO:	Registro de lineas de negocio
NOMBRE CRC:	Lineas de Negocio		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un registro de una linea de negocio.</li> <li>- Obtener un registro de una linea de negocio.</li> <li>- Actualizar un registro de una linea de negocio.</li> <li>- Eliminar un registro de una linea de negocio.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarLineaNegocio</li> <li>- ObtenerLineaNegocio</li> <li>- ActualizarLineaNegocio</li> <li>- EliminarLineaNegocio</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todas las lineas de negocio que posee la empresa para crear los difernetes priyectos.</li> <li>- Una linea de negocio puede tener cero o varios proyectos.</li> </ul>			

Figura 41. Tarjeta CRC\_lineas\_de\_negocio

### Registro perfil del trabajador

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC05	ESCENARIO:	Registro perfil del trabajador
NOMBRE CRC:	Perfiles de Trabajador		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un registro de un perfil de trabajador.</li> <li>- Obtener un registro de un perfil de trabajador.</li> <li>- Actualizar un registro de un perfil de trabajador.</li> <li>- Eliminar un registro de un perfil de trabajador.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarPerfilTrabajador</li> <li>- ObtenerPerfilTrabajador</li> <li>- ActualizarPerfilTrabajador</li> <li>- EliminarPerfilTrabajador</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todas las lineas de negocio que posee la empresa para crear los diferentes proyectos.</li> <li>- Una linea de negocio puede tener cero o varios proyectos.</li> </ul>			

Figura 42. Tarjeta CRC\_perfiles\_de\_trabajador

### Registro de estados de proyectos

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC06	ESCENARIO:	Registro de estados de proyectos
NOMBRE CRC:	Estados de Proyecto		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un registro de un estado de proyecto.</li> <li>- Obtener un registro de un estado de proyecto.</li> <li>- Actualizar un registro de un estado de proyecto.</li> <li>- Eliminar un registro de un estado de proyecto.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarEstadoProyecto</li> <li>- ObtenerEstadoProyecto</li> <li>- ActualizarEstadoProyecto</li> <li>- EliminarEstadoProyecto</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todos los estados que pueden tener los diferentes proyectos de la empresa.</li> <li>- En un estado pueden estar cero o varios proyectos.</li> </ul>			

Figura 43. Tarjeta CRC\_estados\_de\_proyecto

### Registro de contactos

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC07	ESCENARIO:	Registro de contactos
NOMBRE CRC:	Contactos		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un registro de un contacto.</li> <li>- Obtener un registro de un contacto.</li> <li>- Actualizar un registro de un contacto.</li> <li>- Eliminar un registro de un contacto.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarContacto</li> <li>- ObtenerContacto</li> <li>- ActualizarContacto</li> <li>- EliminarContacto</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todos los contactos que interactúan en el ciclo de vida de un proyecto.</li> </ul>			

Figura 44. Tarjeta CRC\_contactos

### 3.5.2 Tarjetas CRC segunda iteración.

## Ingreso cabecera kit.

- Realizar operaciones CRUD.

TARJETA CRC			
NUMERO:	CRC10	ESCENARIO:	Ingreso cabecera kit
NOMBRE CRC:	Cabeceras de Kit		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
- Seleccionar una línea de negocio. - Guardar un registro de una cabecera de un Kit. - Obtener un registro de una cabecera de un Kit. - Actualizar un registro de una cabecera de un Kit. - Eliminar un registro de una cabecera de un Kit. - Calcular totales de un registro de una cabecera de un Kit.	Líneas de Negocio	- ObtenerLineaNegocio - GuardarCabeceraKit - ObtenerCabeceraKit - ActualizarCabeceraKit - EliminarCabeceraKit - CalcularTotales	
OBSERVACIONES:			
- Se registran todas las cabeceras de los Kits que la empresa necesita para los proyectos.			

Figura 45. Tarjeta CRC\_cabeceras\_de\_kit

## Ingreso de materia prima en el kit.

- Agregar, eliminar detalles de materia prima al kit.

TARJETA CRC			
NUMERO:	CRC11	ESCENARIO:	Ingreso de materia prima en el kit
NOMBRE CRC:	Materias Primas en Kit		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
- Obtener un producto del inventario. - Guardar un registro de una Materia Prima. - Obtener un registro de una Materia Prima. - Actualizar un registro de una Materia Prima. - Eliminar un registro de una Materia Prima.		- ObtenerProducto - GuardarMateriaPrima - ObtenerMateriaPrima - ActualizarMateriaPrima - EliminarMateriaPrima	
OBSERVACIONES:			
- Se registran todas las Materias Primas que va a tener el Kit. - En un Kit puede haber cero o mas registros de Materias Primas. - Al momento de obtener el producto se lo realizara con una consulta a la base de datos del ERP.			

Figura 46. Tarjeta CRC\_materia\_prima\_en\_kit

### Registro de mano de obra en el kit.

- Agregar, eliminar detalles de mano de obra al kit.

TARJETA CRC			
NUMERO:	CRC12	ESCENARIO:	Registro de mano de obra en el kit
NOMBRE CRC:	Manos de Obra en Kit		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un perfil de trabajador.</li> <li>- Guardar un registro de una Mano de Obra.</li> <li>- Obtener un registro de una Mano de Obra.</li> <li>- Actualizar un registro de una Mano de Obra.</li> <li>- Eliminar un registro de una Mano de Obra.</li> </ul>	Perfiles de Trabajador	<ul style="list-style-type: none"> <li>- ObtenerPerfilTrabajador</li> <li>- GuardarMateriaPrima</li> <li>- ObtenerMateriaPrima</li> <li>- ActualizarMateriaPrima</li> <li>- EliminarMateriaPrima</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todas las Manos de Obra que va a tener el Kit.</li> <li>- En un Kit puede haber cero o mas registros de Manos de Obra.</li> </ul>			

Figura 47. Tarjeta CRC\_mano\_de\_obra\_en\_kit

### 3.5.3 Tarjetas CRC tercera iteración.

#### Crear una oferta.

- Guardar, modificar, buscar una oferta.



TARJETA CRC			
NUMERO:	CRC20	ESCENARIO:	Crear una oferta
NOMBRE CRC:	Ofertas		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar una linea de negocio.</li> <li>- Guardar un registro de una oferta.</li> <li>- Obtener un registro de una oferta.</li> <li>- Actualizar un registro de una oferta.</li> </ul>	Lineas de Negocio.	<ul style="list-style-type: none"> <li>- ObtenerLineaNegocio</li> <li>- GuardarOferta</li> <li>- ObtenerOferta</li> <li>- ActualizarOferta</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se registran todas las ofertas que la empresa posee.</li> <li>- La oferta tienen que tener un estado inicial y un siguiente estado.</li> </ul>			

Figura 48. Tarjeta CRC\_ofertas

### Ingreso de estados a la oferta.

- Agregar, eliminar detalles de estados a la oferta.

TARJETA CRC			
NUMERO:	CRC21	ESCENARIO:	Ingreso de estados a la oferta
NOMBRE CRC:	Estados en Oferta		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un estado de proyecto.</li> <li>- Guardar un registro de un estado de proyecto.</li> <li>- Eliminar un registro de un estado de proyecto.</li> </ul>	Estados de Proyecto	<ul style="list-style-type: none"> <li>- ObtenerEstadoProyecto</li> <li>- GuardarEstadoOferta</li> <li>- EliminarEstadoOferta</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se agregan los diferentes estados que tiene una oferta.</li> <li>- Una oferta tiene uno o varios estados.</li> </ul>			

Figura 49. Tarjeta CRC\_estados\_en\_oferta

### Ingreso de contactos a la oferta.

- Agregar, eliminar detalles de contactos a la oferta.

TARJETA CRC			
NUMERO:	CRC22	ESCENARIO:	Ingreso de contactos a la oferta
NOMBRE CRC:	Contactos en Oferta		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un registro de contacto</li> <li>- Guardar un registro de un contacto.</li> <li>- Eliminar un registro de un contacto.</li> <li>- Crear un nuevo Contacto</li> </ul>	Contactos  Contactos	<ul style="list-style-type: none"> <li>- ObtenerContacto</li> <li>- GuardarContactoOferta</li> <li>- EliminarContactoOferta</li> <li>- GuardarContacto</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se agrega los contactos que pueden poseer una oferta.</li> </ul>			

Figura 50. Tarjeta CRC\_contactos\_en\_oferta

### 3.5.4 Tarjetas CRC cuarta iteración.

#### Ingreso datos a la cabecera de la cotización.

- Guardar, modificar, buscar una cotización.

TARJETA CRC			
NUMERO:	CRC30	ESCENARIO:	Ingreso datos a la cabecera de la cotización
NOMBRE CRC:	Cabeceras de Cotización		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un registro de una oferta.</li> <li>- Guardar un registro de una cabecera de cotización.</li> <li>- Obtener un registro de una cabecera de cotización.</li> <li>- Actualizar un registro de una cabecera de cotización.</li> <li>- Eliminar un registro de una cabecera de cotización.</li> <li>- Calcular costos de una cabecera de cotizacion.</li> </ul>	Ofertas	<ul style="list-style-type: none"> <li>- ObtenerOferta</li> <li>- GuardarCotizacion</li> <li>- ObtenerCotizacion</li> <li>- ActualizarCotizacion</li> <li>- EliminarCotizacion</li> <li>- CalcularCostos</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se administra las cotizaciones que la empresa emite, para las diferentes ofertas.</li> <li>- Las ofertas que aparecerán para seleccionar en la cotización depende de la configuración de los estado de ofertas.</li> <li>- Una oferta puede tener una o mas cotizaciones.</li> </ul>			

Figura 51. Tarjeta CRC\_cabeceras\_de\_cotizacion

#### Añadir kits a la cotización.

- Agregar, eliminar detalles de kits a la cotización.

TARJETA CRC			
NUMERO:	CRC31	ESCENARIO:	Añadir kits a la cotización
NOMBRE CRC:			Kits en la Cotización
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
- Seleccionar un registro de un Kit de la base de datos - Guardar un detalle de Kit en la cotización. - Obtener un detalle de Kit en la Cotización. - Actualizar un detalle de Kit en la Cotización. - Eliminar un detalle de Kit en la Cotización.  - Modificar Materia Prima del detalle de Kit.  - Modificar Mano de Obra del detalle de Kit.	Cabeceras de Kit	- ObtenerKit  - GuardarKitCotizacion  - ObtenerKitCotizacion  - ActualizarKitCotizacion  - EliminarKitCotizacion  - ModificarMateriaPrimaCotizacion  - ModificarManoObraCotizacion	
OBSERVACIONES:			
- Se administra todos los detalles de Kits que va a tener la Cotización. - Cuando Se modifiquen los datos en la Materia Prima o Mano de Obra del Kit Se cambiarian solo para esa Cotización mas no en el Kit como tal. - Una Cotización puede tener cero o mas Kits.			

Figura 52. Tarjeta CRC\_kits\_en\_la\_cotizacion

### Añadir otros gastos a la cotización.

- Agregar, eliminar detalles de otros gastos en la cotización.

TARJETA CRC			
NUMERO:	CRC32	ESCENARIO:	Añadir otros gastos a la cotización
NOMBRE CRC:			Otros Gastos en la Cotizacion
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
- Guardar un detalle de un gasto en la Cotizacion. - Obtener un detalle de un gasto de la Cotizacion. - Actualizar un detalle de un gasto de la Cotizacion. - Eliminar un detalle de un gasto de la Cotizacion.		- GuardarGastoCotizacion  - ObtenerGastoCotizacion  - ActualizarGastoCotizacion  - EliminarGastoCotizacion	
OBSERVACIONES:			
- Se administran los diferentes tipos de Gastos que puede tener la Cotizacion. - En una Cotizacion puede haber uno o varios Gastos.			

Figura 53. Tarjeta CRC\_otros\_gastos\_en\_la\_cotizacion

### Añadir datos informativos en la cotización.

- Agregar, eliminar detalles de datos informativos en la cotización.

TARJETA CRC			
NUMERO:	CRC33	ESCENARIO:	Añadir datos informativos en la cotizacion
NOMBRE CRC:	Datos Informativos en la Cotizacion		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Guardar un detalle de un Dato Informativo en la Cotizacion.</li> <li>- Obtener un detalle de un Dato Informativo de la Cotizacion.</li> <li>- Actualizar un detalle de un Dato Informativo de la Cotizacion.</li> <li>- Eliminar un detalle de un Dato Informativo de la Cotizacion.</li> </ul>		<ul style="list-style-type: none"> <li>- GuardarDatoInformativo Cotizacion</li> <li>- ObtenerDatoInformativo Cotizacion</li> <li>- ActualizarDatoInformativo Cotizacion</li> <li>- EliminarDatoInformativo Cotizacion</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Se administran los diferentes tipos de Datos Informativos que puede tener la Cotizacion.</li> <li>- En una Cotizacion puede haber uno o varios Datos Informativos.</li> </ul>			

Figura 54. Tarjeta CRC\_datos\_informativos\_en\_la\_cotizacion

### 3.5.5 Tarjetas CRC quinta iteración.

#### Ingreso de datos al proyecto.

- Guardar, modificar, buscar, copiar una cotización.

TARJETA CRC			
NUMERO:	CRC40	ESCENARIO:	Ingreso de datos al proyecto
NOMBRE CRC:	Cabecera de Proyectos		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un registro de una cabecera de Cotización.</li> <li>- Guardar un registro de una cabecera de Proyecto.</li> <li>- Obtener un registro de una cabecera de Proyecto.</li> <li>- Actualizar un registro de una cabecera de Proyecto.</li> <li>- Eliminar un registro de una cabecera de Proyecto.</li> </ul>	Cabeceras de Cotización	<ul style="list-style-type: none"> <li>- ObtenerCotizacion</li> <li>- GuardarProyecto</li> <li>- ObtenerProyecto</li> <li>- ActualizarProyecto</li> <li>- EliminarProyecto</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Un Proyecto puede tener uno o varias Cotizaciones.</li> </ul>			

Figura 55. Tarjeta CRC\_cabeceras\_de\_proyectos

### Ingreso de personas involucradas al proyecto.

- Agregar, eliminar detalles de personas involucradas en el proyecto.

TARJETA CRC			
NUMERO:	CRC41	ESCENARIO:	Ingreso de personas involucradas al proyecto
NOMBRE CRC:	Personas Involucradas en Proyecto		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un registro de un contacto.</li> <li>- Guardar un registro de una persona involucrada.</li> <li>- Obtener un registro de una persona involucrada.</li> <li>- Actualizar un registro de una persona involucrada.</li> <li>- Eliminar un registro de una persona involucrada.</li> </ul>	Contactos	<ul style="list-style-type: none"> <li>- ObtenerContacto</li> <li>- ObtenerPersonaInvolucrada</li> <li>- ObtenerPersonaInvolucrada</li> <li>- ActualizarPersonaInvolucrada</li> <li>- EliminarPersonaInvolucrada</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Un Proyecto puede tener una o mas Personas Involucradas.</li> </ul>			

Figura 56. Tarjeta CRC\_personas\_involucradas\_en\_proyecto

### Ingreso de clientes al proyecto.

- Agregar, eliminar detalles de clientes en el proyecto.

TARJETA CRC			
NUMERO:	CRC42	ESCENARIO:	Ingreso de clientes al proyecto
NOMBRE CRC:	Clientes en Proyecto		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un registro de un cliente.</li> <li>- Guardar un registro de un cliente de Proyecto.</li> <li>- Obtener un registro de un cliente de Proyecto.</li> <li>- Actualizar un registro de un cliente de Proyecto.</li> <li>- Eliminar un registro de un cliente de Proyecto.</li> </ul>		<ul style="list-style-type: none"> <li>- ObtenerCliente</li> <li>- GuardarClienteProyecto</li> <li>- ObtenerClienteProyecto</li> <li>- ActualizarClienteProyecto</li> <li>- EliminarClienteProyecto</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Un Proyecto puede tener uno o varios Clientes.</li> <li>- Los Clientes se consultan de la Base de Datos del ERP.</li> </ul>			

Figura 57. Tarjeta CRC\_clientes\_en\_proyecto

### Registro de otros datos al proyecto.

- Agregar, eliminar detalles de otros datos en el proyecto.

TARJETA CRC			
NUMERO:	CRC43	ESCENARIO:	Registro de otros datos al proyecto
NOMBRE CRC:	Otros Datos en Proyecto		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un registro de un contacto.</li> <li>- Guardar un registro de Otros Datos en el Proyecto.</li> <li>- Obtener un registro de Otros Datos del Proyecto.</li> <li>- Actualizar un registro de Otros Datos del Proyecto.</li> <li>- Eliminar un registro de Otros Datos del Proyecto.</li> </ul>	Contactos	<ul style="list-style-type: none"> <li>- ObtenerContacto</li> <li>- GuardarOtrosDatos.</li> <li>- ObtenerOtrosDatos.</li> <li>- ActualizarOtrosDatos.</li> <li>- EliminarOtrosDatos.</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Un Proyecto puede tener uno o varios Otros Datos.</li> <li>- Los tipos de Otros Datos que existen son: Persona que recibe facturas y Persona que realiza Pagos .</li> </ul>			

Figura 58. Tarjeta CRC\_otros\_datos\_en\_proyecto

### Ingreso de tareas al proyecto.

- Agregar, eliminar detalles de tareas en el proyecto.

TARJETA CRC			
NUMERO:	CRC44	ESCENARIO:	Ingreso de tareas al proyecto
NOMBRE CRC:	Tareas al Proyecto		
RESPONSABILIDADES:	COLABORADORES:	MÉTODOS:	
<ul style="list-style-type: none"> <li>- Seleccionar un Kit del Detalle de la Cotizacion.</li> <li>- Guardar un registro de una tarea en el Proyecto</li> <li>- Obtener un registro de una tarea del Proyecto</li> <li>- Actualizar un registro de una tarea del Proyecto</li> <li>- Eliminar un registro de una tarea del Proyecto</li> </ul>		<ul style="list-style-type: none"> <li>- ObtenerKitDetalleCotizacion</li> <li>- GuardarTarea</li> <li>- ObtenerTarea</li> <li>- ActualizarTarea</li> <li>- EliminarTarea</li> </ul>	
OBSERVACIONES:			
<ul style="list-style-type: none"> <li>- Un Proyecto puede tener uno o varias Tareas.</li> </ul>			

Figura 59. Tarjeta CRC\_tareas\_al\_proyecto

### 3.6 Diagrama Entidad Relación

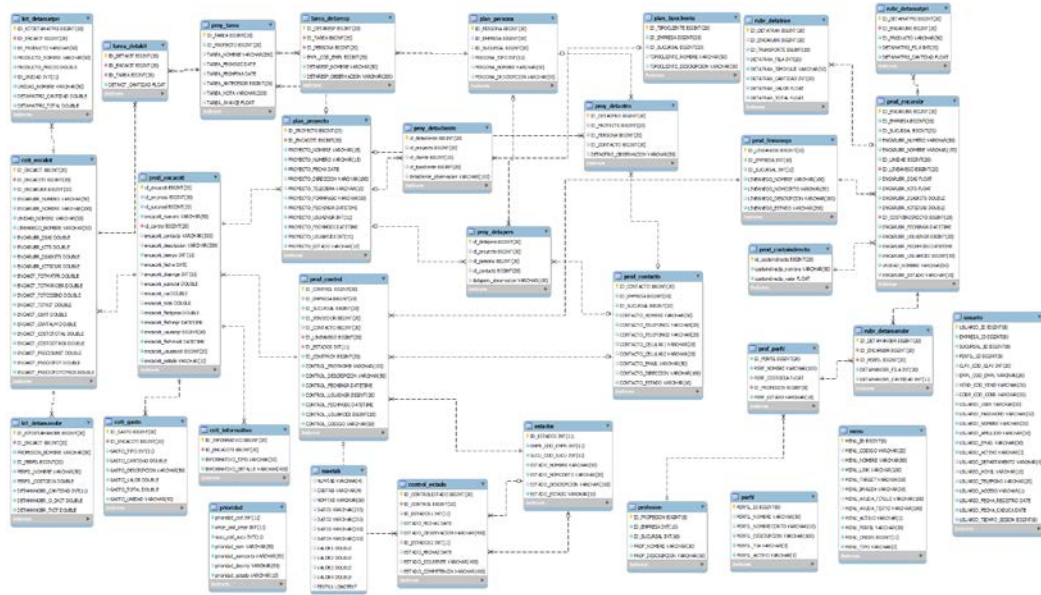


Figura 60. Diagrama entidad relación

## **4 CODIFICACION**

### **4.1 Cliente - Programador**

La metodología XP menciona que el cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada.

Por este motivo la empresa SISCONTI dispone que los días lunes, martes y miércoles los desarrollos y avances del proyecto se los realice en la empresa INCOAYAM para poder tener más contacto con el usuario que en este caso son los trabajadores de la empresa. Mientras que los días jueves, y viernes se lo hará en la empresa SISCONTI para que el programador en este caso el desarrollador de la tesis pueda tener respaldo por el líder del proyecto.

### **4.2 Programación en pareja**

Según la metodología XP, toda la producción de código debe de ser en parejas, sentadas en frente de un solo monitor para tener un diseño de mejor calidad y un código más organizado, teniendo en cuenta que los problemas que se presentan se pueden solucionar más fácilmente y rápidamente.

Debido a que el desarrollo de este proyecto, es individual, este objetivo fue muy difícil de aplicar; lo que se tuvo que implementar fue que el líder del proyecto, ayudo a colaborar con el programador a realizar la programación en pareja en sus tiempos libres.

### **4.3 Integración secuencial**

Para aplicar esta etapa de XP, se debe de saber que antes de aplicar nuevo código al proyecto, se debe garantizar que la última versión halla pasado todas las pruebas,



solo se debe hacer una integración a la vez y se debe de tener claro cuál es la última versión.

Desde el primer momento se llevó un registro de los respaldos de todos los días del último código, el cual su nombre estaba compuesto de la fecha con el proyecto, y el último script de la base de datos.

Estos respaldos eran almacenados en dos partes, un respaldo externo que estaba en el servidor de la empresa SISCONTI y un respaldo local, haciendo que la información esté libre de cualquier eventualidad con respecto al equipo en el que se estaba desarrollando.

Las versiones que se manejaron por el hecho de que existe un solo programador fueron de una sola versión, haciendo que si existe algún error se pueda revisar respaldos anteriores y poder hacer un merge del mismo.

#### **4.4 Integraciones frecuentes**

Los principios fundamentales de la metodología con respecto a este tema, son que se debe hacer integraciones en cada momento como un máximo de tiempo de un día; entre más se tarde en encontrar un problema, resulta más costoso resolverlo, y la programación frecuente evita problemas como el trabajar sobre una clase obsoleta.

Durante el transcurso de este proyecto, no fue aplicado este tema, ya que el grupo de desarrollo se conformaba por una sola persona y él fue el único encargado de modificar la única versión, es decir podía cambiar clases sin que otra parte se vea afectada por el cambio.

#### **4.5 Estándares de programación**

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación

utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

Seguir esta recomendación fue fácil para el desarrollo del proyecto ya que para este caso existe un solo desarrollador, sin embargo lo que se tuvo en cuenta fue que las variables, nombres de métodos sean declaradas con un nombre que se lo suficientemente entendible de acuerdo a su función.

Un ejemplo de lo que no se debe hacer es tener una función que haga el cálculo de una suma de resultados y se declare con el nombre function cal\_sum(), lo correcto debería ser calculo funtion calculo\_totales().

## 4.6 Desarrollo de la aplicación

Para el desarrollo de la aplicación, para cada módulo se consideró el uso de tres archivos principales los cuales se detallan a continuación.

### 4.6.1 Página principal

Este archivo .php es el que contiene todo lo que interpreta el lado del cliente, se conforma en primer lugar por incluir archivos .php que son los que generan el menú iterativo que usa la aplicación.

```

1 <? /***** / ?>
2 <? /* NO MODIFICAR ESTA SECCION*/ ?>
3 <? include_once('./_Modulo.inc.php');?>
4 <? include_once(HEADER_MODULO);?>
5 <? if ($ejecuta) { ?>
6 <? /***** / ?>

```

**Figura 61. Cabecera página principal**

En el segundo bloque se declara todas las funciones javascript que funcionan del lado del cliente y a su vez hacen posible la comunicación mediante ajax con el servidor donde se encuentra la lógica del negocio.

```

8 <script>
9
10     function genera_formulario(opcion,id_profesion){...3 lines }
13
14     function crud_perfil(opcion, id_DGPerfil){...3 lines }
17
18     function crud_profesion(opcion){...6 lines }
24
25     function limpiar(){
26         document.getElementById("perf_nombre").value = '';
27         document.getElementById("perf_costodia").value = 0;
28
29     }
30
31     function consultar_profesion(){
32         if(document.getElementById("id_profesion").value)
33         {
34             xajax_genera_formulario('consultar',xajax.getFormValues("form1"));
35             xajax_consultar_perfil(xajax.getFormValues("form1"));
36         }
37         else
38         {
39             alert("Por favor ingrese una profesion a buscar");
40         }
41     }
42
43 </script>
44

```

**Figura 62. Bloque de scripts página principal**

En el tercer bloque se encuentra todo el código html que interpreta el navegador además de invocar a una función javascript que genera el formulario de acuerdo a la lógica de cada módulo.

```

45 <!--DIBUJA FORMULARIO FILTRO-->
46 <div align="center">
47     <form id="form1" name="form1" action="javascript:void(null);">
48         <table align="center" border="0" cellpadding="2" cellspacing="0" width="75%">
49             <tr>
50                 <td valign="top" align="center">
51                     <div id="divFormularioCabecera"></div>
52                 </td>
53             </tr>
54             <tr>
55                 <td valign="top" align="center">
56                     <div id="divFormularioDetalle"></div>
57                 </td>
58             </tr>
59             <tr>
60                 <td valign="top" align="center">
61                     <div id="divReporte"></div>
62                 </td>
63             </tr>
64         </table>
65     </form>
66 </div>
67 <div id="divGrid" ></div>
68 <script>genera_formulario('nuevo');/*genera_detalle();genera_form_detalle();*/</script>

```

**Figura 63. Estructura HTML página principal**

Por último consta de una sección donde se incluye un archivo .php más el cual hace el papel de un footer en cada módulo del sistema.

```
69 <? /***** / ?>
70 <? /* NO MODIFICAR ESTA SECCION*/ ?>
71 <? } ?>
72 <? include_once(FOOTER_MODULO); ?>
   <? /***** / ?>
```

**Figura 64. Footer página principal**

#### 4.6.2 Archivo común ajax

En el primer bloque se utiliza la función `include_once` propia del php la cual incluye archivos .php los cuales son:

- **config.inc.php:** En este archivo se define todas las constantes que va a utilizar el sistema como directorios de carpetas de imágenes, o de archivos comunes que frecuentemente se utiliza en el código.
- **mysql.inc.php:** Este archivo es el que contiene todas las variables con la cadena de conexión hacia la base del MySQL.
- **informix.inc.php:** Este archivo es el que contiene todas las variables con la cadena de conexión hacia la base de datos Informix del ERP.
- **comun.lib.php:** En esta librería consta de algunas funciones útiles que sean genéricas para poder utilizar dentro de todo el proyecto.
- **Formulario.class.php:** Esta librería tiene todas las funciones que generan código html de una manera genérica.
- **xajax.inc.php:** Es la librería propia de Ajax que utilizaremos dentro de todo el proyecto.

```

2  /* ARCHIVO COMUN PARA LA EJECUCION DEL SERVIDOR AJAX DEL MODULO */
3  /*****/
4  /* NO MODIFICAR */
5  include_once('../Include/config.inc.php');
6  include_once(path(DIR_INCLUDE).'conexiones/mysql.inc.php');
7  include_once(path(DIR_INCLUDE).'conexiones/informix.inc.php');
8  include_once(path(DIR_INCLUDE).'comun.lib.php');
9  include_once(path(DIR_INCLUDE).'Clases/Formulario/Formulario.class.php');
10 require_once (path(DIR_INCLUDE).'Clases/xajax/xajax_core/xajax.inc.php');
11 /*****/

```

**Figura 65. Librerías archivo común**

En la segunda parte de este archivo común se realiza una instancia del servidor Ajax para cada módulo del sistema.

```

12 /* INSTANCIA DEL SERVIDOR AJAX DEL MODULO*/
13 $xajax = new xajax('_Ajax.server.php');
14 $xajax->setCharEncoding('ISO-8859-1');
15 /*****/

```

**Figura 66. Instancia ajax archivo común**

Y por último en el tercer bloque se registra todas las funciones que se van a ejecutar en el servidor de acuerdo a la lógica del negocio de cada uno de los módulos que se van a desarrollar.

```

15 /*****/
16 // FUNCIONES PUBLICAS DEL SERVIDOR AJAX DEL MODULO
17 // Aqui registrar todas las funciones publicas del servidor ajax
18 // Ejemplo,
19 // $xajax->registerFunction("Nombre de la Funcion");
20 /*****/
21 // Funciones de lista de pedido
22 $xajax->registerFunction("genera_formulario");
23 $xajax->registerFunction("gestion_perfil");
24 $xajax->registerFunction("gestion_profesion");
25 $xajax->registerFunction("consultar_perfil");
26
27 /*****/

```

**Figura 67. Registro funciones archivo común**

### 4.6.3 Archivo server

Este archivo consta de tres partes esenciales para su funcionamiento, en el primer bloque se hace la utilización de la función require de php en donde se registra el archivo común ajax mencionado anteriormente.

```

1 <?php
2 require("_Ajax.comun.php"); // No modificar esta linea
3 /*:.....:
4 // S E R V I D O R   A J A X //
5 :.....:
6 /**
7  Herramientas de apoyo
8  */

```

**Figura 68. Require del archivo server**

Para el segundo bloque es el desarrollo de todas las funciones que utiliza el modulo para el desarrollo de la lógica del negocio como tal.

```

10  /*:.....:
11  /* DF01 :: GENERA FORMULARIO PROCESO */
12  /*:.....:
164 function genera_formulario($sAccion='', $sForm='', $id_profesion=0) {...151 lines }
164
164 function gestiona_perfil($opcion = '', $id_DGPerfil = 0 , $sForm = '') {...116 lines }
281
281 function gestiona_profesion($opcion = '', $sForm = '') {...156 lines }
438
438 function consultar_perfil($sForm = '') {...60 lines }
499
499 function genera_grid($aData = null, $aLabel = null, $sTitulo = 'Reporte', $iAncho = '400'
571
572 function consulta_string($sql, $campo, $Conexion, $defecto) {...16 lines }

```

**Figura 69. Funciones del archivo server**

Y en para finalizar en el último bloque se hace el proceso request de las funciones mediante ajax.

```

589 /*:.....:
590 /* PROCESO DE REQUEST DE LAS FUNCIONES MEDIANTE AJAX NO MODIFICAR */
591 $xajax->processRequest();
592 /*:.....:
?>

```

**Figura 70. Request del archivo server**

## 5 PRUEBAS

Mediante la planificación de iteraciones y en base a la especificación de historias de usuario, se crea las pruebas de aceptación, también denominadas pruebas de funcionalidad (documentos formales para determinar la aceptación de funcionalidad que posee un sistema (Historias de Usuario)), las mismas que son constantes y constituyen uno de los pilares básicos de la metodología XP, permitiendo reducir el número de errores e incrementar la calidad del producto; representan una salida del sistema que el cliente espera sea funcional, además de ayudar a realizar un seguimiento del código a emplear, así como también de los cambios y modificaciones que se realizan en la estructura de cada tarea de desarrollo (sin que esto afecte a la funcionalidad de la aplicación), e incluso si se añade nuevas funcionalidades, brindando la posibilidad de reducir el porcentaje de fallos y constituyendo de esta manera una garantía de que la aplicación cumple con los requerimientos establecidos.

Para la documentación formal de las pruebas de aceptación, se procede a la utilización de la siguiente plantilla.

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	
NUMERO CASO DE PRUEBA:	NUMERO HISTORIA DE USUARIO:
NOMBRE CASO DE PRUEBA:	
DESCRIPCION:	
CONDICIONES DE EJECUCION:	
ENTRADAS:	
RESULTADO ESPERADO:	
EVALUACION:	

Figura 71. Plantilla pruebas de aceptación

### 5.1.1 Pruebas de aceptación primera iteración.

Para este módulo se especifica las pruebas de aceptación correspondientes a las historias de usuario pertenecientes a administración del sistema.

- Registro de información de usuarios.
- Registro de prioridades de los proyectos.
- Registro de los perfiles de usuario.
- Registro de las líneas de negocio.
- Registro de perfiles del trabajador.
- Registro de estados del proyecto.
- Registro de contactos.



PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de usuarios		
NUMERO CASO DE PRUEBA:	CP01.01	NUMERO HISTORIA DE USUARIO:	HU01
NOMBRE CASO DE PRUEBA:	Guardar Usuario		
DESCRIPCION:	Se realiza la creacion de usuarios con los datos: Empresa, Perfil, Nombre, Apellido, UserName, Fecha Ingreso, Fecha Caducidad, Email, Estado, Telefono Movil, Telefono Fijo, Tiempo de Sesion.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el ERP tienen que estar creadas las empresas a seleccionar. En el sistema tienen que estar registrados los perfiles.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Usuarios.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema transforma el Nombre y Apellido a Mayusculas.</li> <li>- Al escribir el Nombre de Usuario automaticamente se va añadiendo al Username.</li> <li>- Al escribir el Apellido de Usuario automaticamente se va añadiendo al Username.</li> <li>- Fecha de Ingreso tiene que ser menor a la Fecha de Caducidad.</li> <li>- El Email tienen que ser validado de acuerdo a la estructura del mismo.</li> <li>- El Usuario tiene un solo estado Activo o Inactivo.</li> <li>- El Telefono movil, Telefono Fijo son numéricos.</li> <li>- El sistema emite un mensaje de confirmación de guardado.</li> <li>- El sistema valida que el usuario no este registrado.</li> <li>- El sistema valida los campos obligatorios: Empresa, Perfil, Nombre, Apellido, Usuario, Fecha ingreso, Fecha Caducidad, Email, Usuario Activo.</li> </ul>		
EVALUACION POR EL USUARIO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El Usuario se guarda exitosamente.</li> <li>- Los mensajes se desplagan correctamente en esta prueba</li> </ul>		

**Figura 72. Prueba de aceptación guardar usuario**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de usuarios		
NUMERO CASO DE PRUEBA:	CP01.02	NUMERO HISTORIA DE USUARIO:	HU01
NOMBRE CASO DE PRUEBA:	Buscar Usuario		
DESCRIPCION:	Se realiza la busqueda de usuario por Nombre, Apellido o UserName.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el usuario a buscar.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Usuarios.</li> <li>4. El usuario ingresa el tipo de parametro a buscar.</li> <li>5. El usuario ingresa el parametro a buscar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- Cada vez que se ingrese un caracter el sistema buscara en la base de datos de acuerdo al parametro que se pidio buscar.</li> <li>- El sistema despliega todas las coincidencias en una lista.</li> <li>- Los datos que se desplagan en la lista son Id, Nombre, Apellido, Perfil, Empresa, Estado, Registro, Teléfono.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- Por cada caracter ingresado en el parametro de busqueda se realiza una consulta a la base de datos de acuerdo a toda la cadena ingresada.</li> <li>- El Usuario se Obtiene exitosamente.</li> <li>- Los datos en la lista se desplagan correctamente.</li> </ul>		

**Figura 73. Prueba de aceptación buscar usuario**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de usuarios		
NUMERO CASO DE PRUEBA:	CP01.03	NUMERO HISTORIA DE USUARIO:	HU01
NOMBRE CASO DE PRUEBA:	Editar Usuario		
DESCRIPCION:	Se realiza la edición de la información correspondiente a los Usuarios, los datos que se pueden editar son : Nombre, Apellido, Fecha Ingreso, Fecha Caducidad, Email, Usuario Activo, Teléfono Fijo, Teléfono Movil, Tiempo de Sesion.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el usuario a editar. Se debe primero buscar el Usuario.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Usuarios.</li> <li>4. El usuario busca el registro a editar.</li> <li>5. El usuario da click en el usuario a editar.</li> <li>6. El usuario da click en el boton modificar.</li> <li>7. El usuario ingresa los cambios.</li> <li>8. El usuario da click en el botón Modificar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- Fecha de Ingreso tiene que ser menor a la Fecha de Caducidad.</li> <li>- El Email tienen que ser validado de acuerdo a la estructura del mismo.</li> <li>- El Usuario tiene un solo estado Activo o Inactivo.</li> <li>- El Telefono movil, Telefono Fijo son numéricos.</li> <li>- El sistema emite un mensaje de confirmación de actualizado.</li> <li>- El sistema valida los campos obligatorios: Nombre, Apellido, Fecha Ingreso, Fecha Caducidad, Email, Usuario Activo, Tiempo de Sesión.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El Usuario se actualiza exitosamente.</li> <li>- Los mensajes se desplagan correctamente en esta prueba.</li> </ul>		

**Figura 74. Prueba de aceptación editar usuario**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de usuarios		
NUMERO CASO DE PRUEBA:	CP01.04	NUMERO HISTORIA DE USUARIO:	HU01
NOMBRE CASO DE PRUEBA:	Eliminar Usuario		
DESCRIPCION:	Se realiza el eliminado de usuario: Empresa, Perfil, Nombre, Apellido, UserName, Fecha Ingreso, Fecha Caducidad, Email, Estado, Telefono Movil, Telefono Fijo.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el usuario a eliminar. Se debe primero buscar el Usuario.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Usuarios.</li> <li>4. El usuario busca el registro a eliminar.</li> <li>5. El usuario da click en el boton eliminar.</li> <li>6. El usuario confirma la eliminación.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema realiza un eliminado logico.</li> <li>- El sistema despliega un mensaje de confirmacion de eliminacion.</li> <li>- El sistema despliega un mensaje de que se elimino correctamente el registro.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El Usuario se elimina exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>		

**Figura 75. Prueba de aceptación eliminar usuario**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de usuarios
NUMERO CASO DE PRUEBA:	CP01.05   NUMERO HISTORIA DE USUARIO: HU01
NOMBRE CASO DE PRUEBA:	Resetear Clave
DESCRIPCION:	Se realiza el reseteo de clave de usuario.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Cambiar Clave.</li> <li>3. El usuario ingresa la clave anterior, nueva clave, confirmar clave nueva.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema actualiza la clave del usuario.</li> <li>- El sistema despliega un mensaje de confirmacion de cambio de clave con exito.</li> <li>- El sistema deslogea al usuario para ingresar con la nueva clave.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- El cambio de clave se realizó con exito.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> <li>- El sistema deslogea al usuario.</li> </ul>

**Figura 76. Prueba de aceptación resetear clave**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de prioridades de los proyectos
NUMERO CASO DE PRUEBA:	CP02.01   NUMERO HISTORIA DE USUARIO: HU02
NOMBRE CASO DE PRUEBA:	Guardar Prioridad de Proyecto
DESCRIPCION:	Se realiza la creacion de prioridades de Proyecto con los datos: Nombre, Nombre Corto, Descripcion, Estado).
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Estados de Proyecto.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema transforma el Nombre y Nombre Corto a mayusculas.</li> <li>- El sistema guarda con un Estado True.</li> <li>- El sistema emite un mensaje de confirmacion de guardado.</li> <li>- El sistema valida que el usuario no este registrado de acuerdo al nombre corto.</li> <li>- El sistema valida los campos obligatorios: Nombre, Nombre Corto.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- La Prioridad se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

**Figura 77. Prueba de aceptación guardar prioridad de proyecto**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de prioridades de los proyectos
<b>NUMERO CASO DE PRUEBA:</b>	CP02.02   <b>NUMERO HISTORIA DE USUARIO:</b> HU02
<b>NOMBRE CASO DE PRUEBA:</b>	Buscar Prioridad de Proyecto
<b>DESCRIPCION:</b>	Se realiza la búsqueda de Prioridad de Proyecto por Nombre, o Nombre Corto.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la prioridad a buscar.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Estados de Proyecto. 4. El usuario ingresa el parametro a buscar. 5. El usuario da click en el boton Buscar.
<b>RESULTADO ESPERADO:</b>	- El sistema despliega todas las coincidencias en una lista. - Los datos que se despliegan en la lista son: N°, Nombre, Nombre Corto, Descripción, Estado.
<b>EVALUACION:</b>	- La Prioridad de Proyecto se busca exitosamente. - Los datos en la lista se despliegan correctamente.

**Figura 78. Prueba de aceptación buscar prioridad de proyecto**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de prioridades de los proyectos
<b>NUMERO CASO DE PRUEBA:</b>	CP02.03   <b>NUMERO HISTORIA DE USUARIO:</b> HU02
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Prioridad de Proyecto
<b>DESCRIPCION:</b>	Se realiza la edición de la información correspondiente a la Prioridad de Proyecto, los datos que se pueden editar son : Nombre, Nombre Corto, Descripción.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la Prioridad de Proyecto a editar. Se debe primero buscar la Prioridad de Proyecto.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Prioridades de Proyecto. 4. El usuario busca el registro a editar. 5. El usuario da click en el boton Modificar. 6. El usuario hace los cambios en los campos. 7. El usuario da click en el boton Modificar.
<b>RESULTADO ESPERADO:</b>	- El sistema transforma el nombre corto a mayusculas. - El sistema emite un mensaje de confirmacion de actualizacion. - El sistema valida que el usuario no este registrado de acuerdo al nombre corto. - El sistema valida los campos obligatorios: Nombre, Nombre Corto.
<b>EVALUACION:</b>	- Las validaciones se efectuan correctamente en esta prueba. - El Usuario se actualiza exitosamente. - Los mensajes se despliegan correctamente en esta prueba.

**Figura 79. Prueba de aceptación editar prioridad de proyecto**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de prioridades de los proyectos
NUMERO CASO DE PRUEBA:	CP02.04   NUMERO HISTORIA DE USUARIO: HU02
NOMBRE CASO DE PRUEBA:	Eliminar Prioridad de Proyecto
DESCRIPCION:	Se realiza el eliminado de Prioridad de Proyecto con los datos: Nombre, Nombre Corto, Descripcion, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado la Prioridad a eliminar. Se debe primero buscar la Prioridad.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Prioridades de Proyecto.</li> <li>4. El usuario busca el registro a eliminar.</li> <li>5. El usuario da click en el boton eliminar.</li> <li>6. El usuario confirma la eliminación.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema realiza un eliminado logico.</li> <li>- El sistema despliega un mensaje de confirmacion de eliminacion.</li> <li>- El sistema despliega un mensaje de que se elimino correctamente el registro.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- La Prioridad se elimina exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

Figura 80. Prueba de aceptación eliminar prioridad de proyecto

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de perfiles de usuario
NUMERO CASO DE PRUEBA:	CP03.01   NUMERO HISTORIA DE USUARIO: HU03
NOMBRE CASO DE PRUEBA:	Guardar Perfil
DESCRIPCION:	Se realiza la creacion de Perfil de Usuario con los datos: Nombre, Nombre Corto, Descripcion, Permiso, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Perfiles.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema transforma el nombre corto a mayusculas.</li> <li>- El sistema guarda con un Estado True.</li> <li>- El sistema emite un mensaje de confirmacion de guardado.</li> <li>- El sistema valida que el usuario no este registrado de acuerdo al Nombre.</li> <li>- El sistema valida los campos obligatorios: Nombre, Nombre Corto, Permiso.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El Perfil se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba de esta prueba.</li> </ul>

Figura 81. Prueba de aceptación guardar perfil

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de perfiles de usuario
<b>NUMERO CASO DE PRUEBA:</b>	CP03.02   <b>NUMERO HISTORIA DE USUARIO:</b> HU03
<b>NOMBRE CASO DE PRUEBA:</b>	Buscar Perfil
<b>DESCRIPCION:</b>	Se realiza la búsqueda de Perfil por Nombre, o Nombre Corto.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Perfil a buscar.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Perfiles. 4. El usuario ingresa el parametro a buscar. 5. El usuario da click en el boton Buscar.
<b>RESULTADO ESPERADO:</b>	- El sistema despliega todas las coincidencias en una lista. - Los datos que se despliegan en la lista son: N°, Nombre, Nombre Corto, Descripción, Estado.
<b>EVALUACION:</b>	- El Perfil se busca exitosamente. - Los datos en la lista se despliegan correctamente.

**Figura 82. Prueba de aceptación buscar perfil**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de perfiles de usuario
<b>NUMERO CASO DE PRUEBA:</b>	CP03.03   <b>NUMERO HISTORIA DE USUARIO:</b> HU03
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Perfil
<b>DESCRIPCION:</b>	Se realiza la edición de la información correspondiente al Perfil, los datos que se pueden editar son : Nombre, Nombre Corto, Descripción, Permiso.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Perfil a editar. Se debe primero buscar el Perfil.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Perfiles. 4. El usuario busca el registro a editar. 5. El usuario da click en el boton Modificar. 6. El usuario hace los cambios en los campos. 7. El usuario da click en el boton Modificar.
<b>RESULTADO ESPERADO:</b>	- El sistema transforma el nombre corto a mayusculas. - El sistema emite un mensaje de confirmacion de actualizacion. - El sistema valida que el Perfil no este registrado de acuerdo al Nombre. - El sistema valida los campos obligatorios: Nombre, Nombre Corto, Permiso.
<b>EVALUACION:</b>	- Las validaciones se efectuan correctamente en esta prueba. - El Perfil se actualiza exitosamente. - Los mensajes se despliegan correctamente en esta prueba.

**Figura 83. Prueba de aceptación editar perfil**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de perfiles de usuario
NUMERO CASO DE PRUEBA:	CP03.04   NUMERO HISTORIA DE USUARIO:   HU03
NOMBRE CASO DE PRUEBA:	Eliminar Perfil
DESCRIPCION:	Se realiza el eliminado de Perfil con los datos: Nombre, Nombre Corto, Descripcion, Permiso, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Perfil a eliminar. Se debe primero buscar el Perfil.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Perfiles.</li> <li>4. El usuario busca el registro a eliminar.</li> <li>5. El usuario da click en el boton eliminar.</li> <li>6. El usuario confirma la eliminación.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema realiza un eliminado logico.</li> <li>- El sistema despliega un mensaje de confirmacion de eliminacion.</li> <li>- El sistema despliega un mensaje de que se elimino correctamente el registro.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- El Perfil se elimina exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

**Figura 84. Prueba de aceptación eliminar perfil**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de lineas de negocio
NUMERO CASO DE PRUEBA:	CP04.01   NUMERO HISTORIA DE USUARIO:   HU04
NOMBRE CASO DE PRUEBA:	Guardar Linea
DESCRIPCION:	Se realiza la creacion de Linea con los datos: Nombre, Nombre Corto, Descripcion, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Lineas de Negocio.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema transforma el nombre corto a mayusculas.</li> <li>- El sistema guarda con un Estado True.</li> <li>- El sistema emite un mensaje de confirmacion de guardado.</li> <li>- El sistema valida que la Linea no este registrado de acuerdo al Nombre Corto.</li> <li>- El sistema valida los campos obligatorios: Nombre, Nombre Corto.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- La Linea se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

**Figura 85. Prueba de aceptación guardar línea**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de líneas de negocio
<b>NUMERO CASO DE PRUEBA:</b>	CP04.02   <b>NUMERO HISTORIA DE USUARIO:</b> HU04
<b>NOMBRE CASO DE PRUEBA:</b>	Buscar Línea
<b>DESCRIPCION:</b>	Se realiza la búsqueda de Línea por Nombre, o Nombre Corto.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado la Línea a buscar.
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Lineas de Negocio.</li> <li>4. El usuario ingresa el parametro a buscar.</li> <li>5. El usuario da click en el boton Buscar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	- El sistema despliega todas las coincidencias en una lista. - Los datos que se despliegan en la lista son: N°, Nombre, Nombre Corto, Descripción, Estado.
<b>EVALUACION:</b>	- La Línea se busca exitosamente. - Los datos en la lista se despliegan correctamente.

**Figura 86. Prueba de aceptación buscar línea**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de líneas de negocio
<b>NUMERO CASO DE PRUEBA:</b>	CP04.03   <b>NUMERO HISTORIA DE USUARIO:</b> HU04
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Línea
<b>DESCRIPCION:</b>	Se realiza la edición de la información correspondiente a la Línea, los datos que se pueden editar son : Nombre, Nombre Corto, Descripción.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la Línea a editar. Se debe primero buscar la Línea.
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Lineas de Negocio.</li> <li>4. El usuario busca el registro a editar.</li> <li>5. El usuario da click en el boton Modificar.</li> <li>6. El usuario hace los cambios en los campos.</li> <li>7. El usuario da click en el boton Modificar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	- El sistema transforma el nombre corto a mayúsculas. - El sistema emite un mensaje de confirmación de actualización. - El sistema valida que la Línea no este registrada de acuerdo al Nombre Corto. - El sistema valida los campos obligatorios: Nombre, Nombre Corto.
<b>EVALUACION:</b>	- Las validaciones se efectuan correctamente en esta prueba. - La Línea se actualiza exitosamente. - Los mensajes se despliegan correctamente en esta prueba.

**Figura 87. Prueba de aceptación editar línea**



<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de lineas de negocio
<b>NUMERO CASO DE PRUEBA:</b>	CP04.04 <b>NUMERO HISTORIA DE USUARIO:</b> HU04
<b>NOMBRE CASO DE PRUEBA:</b>	Eliminar Linea
<b>DESCRIPCION:</b>	Se realiza el eliminado de Linea con los datos: Nombre, Nombre Corto, Descripcion, Estado.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la Linea a eliminar. Se debe primero buscar la Linea.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Lineas de Negocio. 4. El usuario busca el registro a eliminar. 5. El usuario da click en el boton eliminar.
<b>RESULTADO ESPERADO:</b>	- El sistema realiza un eliminado logico. - El sistema despliega un mensaje de que se elimino correctamente el registro.
<b>EVALUACION:</b>	- La Linea se elimina exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 88. Prueba de aceptación eliminar línea**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro perfil del trabajador
<b>NUMERO CASO DE PRUEBA:</b>	CP05.01 <b>NUMERO HISTORIA DE USUARIO:</b> HU05
<b>NOMBRE CASO DE PRUEBA:</b>	Guardar Perfil de Trabajador
<b>DESCRIPCION:</b>	Se realiza la creacion de Perfil de Trabajador con los datos: Nombre Cargo, Descripcion, Nombre Perfil, Costo por Dia, Estado.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Trabajador. 4. El usuario da click en el boton Nuevo. 5. El usuario llena los datos de la pantalla. 6. El usuario da click en agregar Perfil. 7. El usuario da click en el boton Guardar.
<b>RESULTADO ESPERADO:</b>	- El boton nuevo limpia todos los datos de la pantalla. - El sistema transforma el Nombre Cargo y Nombre Perfil a mayusculas. - El sistema valida que el Costo por Dia sea solo numeros. - El sistema guarda con un Estado True. - El sistema emite un mensaje de confirmacion de guardado. - El sistema valida que el Perfil no este registrado de acuerdo al Nombre Cargo. - El sistema valida los campos obligatorios: Nombre Cargo , Nombre Perfil y Costo por Dia.
<b>EVALUACION:</b>	- El boton nuevo limpia todos los campos. - Las validaciones se efectuan correctamente en esta prueba. - El Perfil de Trabajador se guarda exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 89. Prueba de aceptación guardar perfil de trabajador**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro perfil del trabajador
<b>NUMERO CASO DE PRUEBA:</b>	CP05.02   <b>NUMERO HISTORIA DE USUARIO:</b> HU05
<b>NOMBRE CASO DE PRUEBA:</b>	Buscar Perfil Trabajador
<b>DESCRIPCION:</b>	Se realiza la búsqueda de Perfil de Trabajador por Nombre del Cargo.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Perfil de Trabajador a buscar.
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Trabajador.</li> <li>4. El usuario escoje el Trabajador a buscar.</li> <li>5. El usuario da click en el boton Buscar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	- El sistema despliega todos los Perfiles del Trabajador en una lista. - Los datos que se desplegan en la lista son: N°, Nombre Perfil, Costo por Dia, Estado.
<b>EVALUACION:</b>	- El Perfil se busca exitosamente. - Los datos en la lista se desplegan correctamente.

**Figura 90. Prueba de aceptación buscar perfil trabajador**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro perfil del trabajador
<b>NUMERO CASO DE PRUEBA:</b>	CP05.03   <b>NUMERO HISTORIA DE USUARIO:</b> HU05
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Perfil Trabajador
<b>DESCRIPCION:</b>	Se realiza la edición de la información correspondiente al Perfil del Trabajador los datos que se pueden editar son : Nombre Cargo, Descripción, Nombre Perfil, Costo por Dia.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Perfil de Trabajador a editar. Se debe primero buscar el Perfil de Trabajador.
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Trabajador.</li> <li>4. El usuario busca el registro a editar.</li> <li>5. El usuario da click en el boton Modificar.</li> <li>6. El usuario hace los cambios en los campos.</li> <li>7. El usuario da click en el boton Actualizar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	- El sistema transforma el Nombre Cargo y Nombre Perfil a mayusculas. - El sistema valida que el Costo por Dia sea solo numeros. - El sistema emite un mensaje de confirmación de actualizado. - El sistema valida que el Perfil no este registrado de acuerdo al Nombre Cargo. - El sistema valida los campos obligatorios: Nombre Cargo , Nombre Perfil y Costo por Dia.
<b>EVALUACION:</b>	- Las validaciones se efectuan correctamente en esta prueba. - El Perfil de Trabajador se guarda exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 91. Prueba de aceptación editar perfil trabajador**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro perfil del trabajador
NUMERO CASO DE PRUEBA:	CP05.04   NUMERO HISTORIA DE USUARIO:   HU05
NOMBRE CASO DE PRUEBA:	Eliminar Perfil de Trabajador
DESCRIPCION:	Se realiza la eliminacion de Perfil de Trabajador con los datos: Nombre Cargo, Descripcion, Nombre Perfil, Costo por Dia, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Perfil de Trabajador a eliminar. Se debe primero buscar la el Perfil de Trabajador a eliminar.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Trabajador. 4. El usuario busca el registro a eliminar. 5. El usuario da click en el boton eliminar. 6. El usuario confirma la eliminación.
RESULTADO ESPERADO:	- El sistema realiza un eliminado logico. - El sistema despliega un mensaje de que se elimino correctamente el registro.
EVALUACION:	- El Perfil de Trabajador se elimina exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 92. Prueba de aceptación eliminar perfil de trabajador**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de estados de proyectos
NUMERO CASO DE PRUEBA:	CP06.01   NUMERO HISTORIA DE USUARIO:   HU06
NOMBRE CASO DE PRUEBA:	Guardar Estado
DESCRIPCION:	Se realiza la creacion de Estado con los datos: Nombre, Nombre Corto, Descripcion, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Estados de Proyecto. 4. El usuario da click en el boton Nuevo. 5. El usuario llena los datos de la pantalla. 6. El usuario da click en el boton Guardar.
RESULTADO ESPERADO:	- El boton nuevo limpia todos los datos de la pantalla. - El sistema transforma el Nombre y Nombre Corto a mayusculas. - El sistema guarda con un Estado True. - El sistema emite un mensaje de confirmacion de guardado. - El sistema valida que el Estado no este registrado de acuerdo al Nombre. - El sistema valida los campos obligatorios: Nombre y Nombre Corto.
EVALUACION:	- El boton nuevo limpia todos los campos. - Las validaciones se efectuan correctamente en esta prueba. - El Estado se guarda exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 93. Prueba de aceptación guardar estado**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de estados de proyectos
<b>NUMERO CASO DE PRUEBA:</b>	CP06.02   <b>NUMERO HISTORIA DE USUARIO:</b>   HU06
<b>NOMBRE CASO DE PRUEBA:</b>	Buscar Estado
<b>DESCRIPCION:</b>	Se realiza la búsqueda del Estado por Nombre o Nombre Corto.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Estado a buscar.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Estados de Proyecto. 4. El usuario ingresa el parametro a buscar. 5. El usuario da click en el boton Buscar.
<b>RESULTADO ESPERADO:</b>	- El sistema despliega todas las coincidencias en una lista. - Los datos que se despliegan en la lista son: N°, Nombre, Nombre Corto, Descripción, Estado.
<b>EVALUACION:</b>	- El Estado se busca exitosamente. - Los datos en la lista se despliegan correctamente.

**Figura 94. Prueba de aceptación buscar estado**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de estados de proyectos
<b>NUMERO CASO DE PRUEBA:</b>	CP06.03   <b>NUMERO HISTORIA DE USUARIO:</b>   HU06
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Estado
<b>DESCRIPCION:</b>	Se realiza la edición de la información correspondiente al Estado los datos que se pueden editar son : Nombre, Nombre Corto, Descripción.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Estado a editar. Se debe primero buscar el Estado.
<b>ENTRADAS:</b>	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Estados de Proyecto 4. El usuario busca el registro a editar. 5. El usuario da click en el boton Modificar. 6. El usuario hace los cambios en los campos. 7. El usuario da click en el boton Modificar.
<b>RESULTADO ESPERADO:</b>	- El sistema transforma el Nombre y Nombre Corto a mayusculas. - El sistema emite un mensaje de confirmacion de guardado. - El sistema valida que el Estado no este registrado de acuerdo al Nombre. - El sistema valida los campos obligatorios: Nombre y Nombre Corto.
<b>EVALUACION:</b>	- Las validaciones se efectuan correctamente en esta prueba. - El Estado se actualiza exitosamente. - Los mensajes se despliegan correctamente en esta prueba.

**Figura 95. Prueba de aceptación editar estado**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de estados de proyectos		
NUMERO CASO DE PRUEBA:	CP06.04	NUMERO HISTORIA DE USUARIO:	HU06
NOMBRE CASO DE PRUEBA:	Eliminar Estado		
DESCRIPCION:	Se realiza el eliminado del Estado con los datos: Nombre, Nombre Corto, Descripcion, Estado.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Estado a eliminar. Se debe primero buscar el Estado.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Estados de Proyecto.</li> <li>4. El usuario busca el registro a eliminar.</li> <li>5. El usuario da click en el boton eliminar.</li> <li>6. El usuario confirma la eliminación.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema realiza un eliminado logico.</li> <li>- El sistema despliega un mensaje de confirmación de eliminacion.</li> <li>- El sistema despliega un mensaje de que se elimino correctamente el registro.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El Estado se elimina exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>		

**Figura 96. Prueba de aceptación eliminar estado**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de contactos		
NUMERO CASO DE PRUEBA:	CP07.01	NUMERO HISTORIA DE USUARIO:	HU07
NOMBRE CASO DE PRUEBA:	Guardar Contacto		
DESCRIPCION:	Se realiza la creación de Contacto con los datos: Nombre, Telefono1, Telefono2, Celular1, Celular2, Email, Direccion, Estado.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Administracion.</li> <li>3. El usuario da click en el boton Contactos.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema transforma el Nombre y Direccion a mayusculas.</li> <li>- El sistema transforma el Email a minusculas.</li> <li>- El sistema guarda con un Estado True.</li> <li>- El sistema emite un mensaje de confirmación de guardado.</li> <li>- El sistema valida los campos obligatorios: Nombre, Telefono1 y Celular1.</li> <li>- El sistema valida que los campos Telefono1, Telefono2, Celular1, Celular2 sean solo numeros.</li> <li>- El Email tienen que ser validado de acuerdo a la estructura del mismo.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El Contacto se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>		

**Figura 97. Prueba de aceptación guardar contacto**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de contactos
NUMERO CASO DE PRUEBA:	CP07.02   NUMERO HISTORIA DE USUARIO: HU07
NOMBRE CASO DE PRUEBA:	Buscar Contacto
DESCRIPCION:	Se realiza la búsqueda del Contacto por Nombre.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Contacto a buscar.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Contactos. 4. El usuario ingresa el parametro a buscar. 5. El usuario da click en el boton Buscar.
RESULTADO ESPERADO:	- El sistema despliega todas las coincidencias en una lista. - Los datos que se despliegan en la lista son: N°, Nombre, Telefono1, Telefono2, Celular1, Celular2, Direccion, Email, Estado.
EVALUACION:	- El Contacto se busca exitosamente. - Los datos en la lista se despliegan correctamente.

**Figura 98. Prueba de aceptación buscar contacto**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de contactos
NUMERO CASO DE PRUEBA:	CP07.03   NUMERO HISTORIA DE USUARIO: HU07
NOMBRE CASO DE PRUEBA:	Editar Contacto
DESCRIPCION:	Se realiza la edición de Contacto con los datos: Nombre, Telefono1, Telefono2, Celular1, Celular2, Email, Direccion.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Contacto a editar. Se debe primero buscar el Contacto.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Contactos. 4. El usuario busca el registro a editar. 5. El usuario da click en el boton Modificar. 6. El usuario hace los cambios en los campos. 7. El usuario da click en el boton Modificar.
RESULTADO ESPERADO:	- El sistema transforma el Nombre a mayusculas. - El sistema emite un mensaje de confirmación de guardado. - El sistema valida los campos obligatorios: Nombre, Telefono1 y Celular1. - El sistema valida que los campos Telefono1, Telefono2, Celular1, Celular2 sean solo numeros. - El Email tienen que ser validado de acuerdo a la estructura del mismo.
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - El Contacto se actualiza exitosamente. - Los mensajes se despliegan correctamente en esta prueba.

**Figura 99. Prueba de aceptación editar contacto**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de contactos
NUMERO CASO DE PRUEBA:	CP07.04
NUMERO HISTORIA DE USUARIO:	HU07
NOMBRE CASO DE PRUEBA:	Eliminar Contacto
DESCRIPCION:	Se realiza el eliminado del Contacto con los datos: Nombre, Telefono1, Telefono2, Celular1, Celular2, Email, Direccion, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Contacto a eliminar. Se debe primero buscar el Contacto.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Administracion. 3. El usuario da click en el boton Contactos. 4. El usuario busca el registro a eliminar. 5. El usuario da click en el boton eliminar.
RESULTADO ESPERADO:	- El sistema realiza un eliminado logico. - El sistema despliega un mensaje de confirmacion de eliminacion. - El sistema despliega un mensaje de que se elimino correctamente el registro.
EVALUACION:	- El Contacto se elimina exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 100. Prueba de aceptación eliminar contacto**

### 5.1.2 Pruebas de aceptación segunda iteración.

Para este módulo se especifica las pruebas de aceptación correspondientes a las historias de usuario pertenecientes a mantenimiento de kits.

- Ingreso de la cabecera del kit.
- Ingreso de materia prima al kit.
- Ingreso de mano de obra al kit.

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso cabecera kit		
NUMERO CASO DE PRUEBA:	CP10.01	NUMERO HISTORIA DE USUARIO:	HU10
NOMBRE CASO DE PRUEBA:	Guardar Kit		
DESCRIPCION:	Se realiza la creacion del Kit con los datos: Codigo, Nombre, Unidad, Linea de Negocio, Dias, Kits, Dias/Kit, Kits/Dia, Costo Indirecto.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. Tienen que estar registradas las unidades en la Base de Datos del ERP. Tienen que estar registradas las Lineas de Negocio. Tienen que estar ingresado el valor del Costo Indirecto en el Sistema.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Kits.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- E sistema genera un codigo automatico basado en el formato "KIT." + LINEA DE NEGOCIO + SECUENCIAL DE TRES DIGITOS.</li> <li>- El sistema transforma el Nombre a mayusculas.</li> <li>- El campo Dias/Kit el sistema calcula de acuerdo a los Dias Ingresados / para el numero de Kits ingresados.</li> <li>- El campo Kits/Dia el sistema calcula de acuerdo a los Kits ingresados / para el numero de Dias ingresados.</li> <li>- El sistema calcula el Total del Kit = Total Materia Prima + Total Mano de Obra + Costo Indirecto.</li> <li>- El sistema emite un mensaje de confirmación de guardado.</li> <li>- El sistema valida los campos obligatorios: Nombre, Unidad, Linea de Negocio, Dias, Kits.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba en esta prueba.</li> <li>- Los Calculos considerados en esta Prueba de Aceptación se efectúan perfectamente.</li> <li>- La Cabecera de Kit se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes en esta prueba.</li> </ul>		

**Figura 101. Prueba de aceptación guardar kit**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso cabecera kit		
NUMERO CASO DE PRUEBA:	CP10.02	NUMERO HISTORIA DE USUARIO:	HU10
NOMBRE CASO DE PRUEBA:	Busqueda Kit		
DESCRIPCION:	Se realiza la busqueda del Kit por Nombre o Codigo.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Kit a buscar.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Kits.</li> <li>3. El usuario ingresa el parametro a buscar.</li> <li>4. El usuario da click en el boton Buscar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema despliega todas las coincidencias en una lista.</li> <li>- Los datos que se desplegan en la lista son: Codigo, Nombre, Unidad.</li> <li>- El sistema al seleccionar el Kit a buscar carga en la pantalla los datos del Kit.</li> <li>- El sistema ejecuta automaticamente el Caso de Prueba (Agregar detalle Materia Prima).</li> <li>- El sistema ejecuta automaticamente el Caso de Prueba (Agregar detalle Mano de Obra).</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El Kit se busca exitosamente.</li> <li>- Los Casos de Prueba se efectuan correctamente.</li> <li>- Los datos en la lista se desplegan correctamente.</li> </ul>		

**Figura 102. Prueba de aceptación búsqueda kit**



PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso cabecera kit		
NUMERO CASO DE PRUEBA:	CP10.03	NUMERO HISTORIA DE USUARIO:	HU10
NOMBRE CASO DE PRUEBA:	Editar Kit		
DESCRIPCION:	Se realiza la edición del Kit con los datos: Nombre, Dias, Kits, Rendimiento.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Kit a editar. Se debe primero buscar el Kit.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Kits.</li> <li>3. El usuario busca el registro a editar.</li> <li>4. El usuario hace los cambios en los campos.</li> <li>5. El usuario da click en el boton Actualizar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema transforma el Nombre a mayusculas.</li> <li>- El campo Dias/Kit el sistema calcula de acuerdo a los Dias ingresados / para el numero de Kits ingresados.</li> <li>- El campo Kits/Dia el sistema calcula de acuerdo a los Kits ingresados / para el numero de Dias ingresados.</li> <li>- El sistema calcula el Total del Kit = Total Materia Prima + Total Mano de Obra + Costo Indirecto.</li> <li>- El sistema emite un mensaje de confirmacion de actualizado.</li> <li>- El sistema valida los campos obligatorios: Nombre, Dias, Kits.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- Los Calculos se efectuan perfectamente.</li> <li>- El Kit se actualiza exitosamente.</li> <li>- El Sistema emite todos los mensajes en esta prueba.</li> </ul>		

Figura 103. Prueba de aceptación editar kit

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso cabecera kit		
NUMERO CASO DE PRUEBA:	CP10.04	NUMERO HISTORIA DE USUARIO:	HU10
NOMBRE CASO DE PRUEBA:	Eliminar Kit		
DESCRIPCION:	Se realiza el eliminado del Kit con los datos:Codigo,Nombre, Unidad, Linea de Negocio, Dias, Kits, Rendimiento, Costo Indirecto.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrado el Kit a eliminar. Se debe primero buscar el Kit.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Kits.</li> <li>3. El usuario busca el registro a eliminar.</li> <li>4. El usuario da click en el boton eliminar.</li> <li>5. El usuario confirma la eliminación.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema realiza un eliminado logico.</li> <li>- El sistema despliega un mensaje de confirmacion de eliminación.</li> <li>- El sistema despliega un mensaje de que se eliminó correctamente el registro.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El Kit se elimina exitosamente.</li> <li>- El Sistema emite todos los mensajes en esta prueba.</li> </ul>		

Figura 104. Prueba de aceptación eliminar kit

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de materia prima en el kit		
NUMERO CASO DE PRUEBA:	CP11.01	NUMERO HISTORIA DE USUARIO:	HU11
NOMBRE CASO DE PRUEBA:	Agregar detalle Materia Prima		
DESCRIPCION:	Se realiza el agregado de detalle de Materia Prima con los Datos: Codigo, Nombre, Unidad, Cantidad, Costo, Total.		
CONDICIONES DE EJECUCION:	Tienen que estar registrados los productos (Codigo, Nombre, Unidad, Costo) en la Base de Datos del ERP. El usuario ejecuta el Caso de Prueba(Guardar Kit).		
ENTRADAS:	1. El usuario busca el Producto a agregar. 2. El usuario ingresa los datos en la pantalla. 3. El usuario da click en el boton Agregar.		
RESULTADO ESPERADO:	- Cada que el usuario ingrese un caracter en el campo producto, automaticamente se buscara por Nombre o Codigo en la Base de Datos del sistema. - El sistema calcula el Total del detalle de la Materia Prima = Costo * Cantidad. - El sistema valida la Cantidad solo ingreso de numeros. - El sistema realiza calculos dinamicos es decir si en la Base de Datos del ERP cambia el Costo del Producto el total del detalle cambia automaticamente. - El sistema agrega el detalle en un Grid. - El sistema valida los campos obligatorios: Codigo, Nombre, Unidad, Costo. - El sistema calcula el Total de la Materia Prima en el Kit = Suma del total de los detalles.		
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - Los Calculos se efectuan perfectamente. - La Materia Prima se agrega exitosamente. - El Sistema emite todos los mensajes en esta prueba.		

**Figura 105. Prueba de aceptación agregar detalle materia prima**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de materia prima en el kit		
NUMERO CASO DE PRUEBA:	CP11.02	NUMERO HISTORIA DE USUARIO:	HU11
NOMBRE CASO DE PRUEBA:	Editar detalle Materia Prima		
DESCRIPCION:	Se realiza el modificado de detalle de Materia Prima con los Datos: Cantidad.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrada la Materia Prima a editar. El usuario ejecuta el Caso de Prueba(Buscar Kit).		
ENTRADAS:	1. El usuario da click en el boton Modificar del detalle de Materia Prima. 2. El usuario realiza los cambios en los campos. 3. El usuario da click en el boton Modificar.		
RESULTADO ESPERADO:	- El sistema calcula El Total del detalle de la Materia Prima = Costo * Cantidad. - El sistema valida la Cantidad solo ingreso de numeros. - El sistema realiza calculos dinamicos es decir si en la Base de Datos del ERP cambia El Costo del Producto El Total del detalle cambia automaticamente. - El sistema modifica el detalle con los nuevos Datos en el Grid. - El sistema valida los campos obligatorios: Cantidad.		
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - Los Calculos se efectuan perfectamente. - El detalle de Materia Prima se actualiza exitosamente. - El Sistema emite todos los mensajes en esta prueba.		

**Figura 106. Prueba de aceptación editar detalle materia prima**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de materia prima en el kit		
NUMERO CASO DE PRUEBA:	CP11.03	NUMERO HISTORIA DE USUARIO:	HU11
NOMBRE CASO DE PRUEBA:	Eliminar detalle Materia Prima		
DESCRIPCION:	Se realiza el eliminado de detalle de Materia Prima con los Datos: Codigo, Nombre, Unidad, Cantidad, Costo, Total.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba(Agregar detalle Materia Prima).		
ENTRADAS:	1. El usuario da click en el boton eliminar.		
RESULTADO ESPERADO:	- El sistema realiza un eliminado físico. - El sistema elimina el detalle del Grid.		
EVALUACION:	- El detalle de Materia Prima se elimina exitosamente. - El Sistema emite todos los mensajes en esta prueba.		

**Figura 107. Prueba de aceptación eliminar detalle materia prima**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Registro de mano de obra en el kit		
NUMERO CASO DE PRUEBA:	CP12.01	NUMERO HISTORIA DE USUARIO:	HU12
NOMBRE CASO DE PRUEBA:	Agregar detalle Mano de Obra		
DESCRIPCION:	Se realiza el agregado de detalle de Mano de Obra con los Datos: Profesion, Perfil, Costo, Cantidad, O-D/KIT, T/KIT.		
CONDICIONES DE EJECUCION:	Tienen que estar registrados los Perfiles de Trabajador. El usuario ejecuta el Caso de Prueba(Guardar Kit).		
ENTRADAS:	1. El usuario escoge el Cargo. 2. El usuario escoge el Perfil. 3. El usuario ingresa la Cantidad. 4. El usuario da click en el boton Agregar.		
RESULTADO ESPERADO:	- Cuando el usuario escoga un Cargo el sistema cargará en el combo box los Perfiles que tenga. - El sistema calcula el campo O-D/KIT del detalle de la Mano de Obra = Cantidad * el valor del campo "Dias/Kit" del Kit. - El sistema calcula el campo T/KIT. del detalle de la Mano de Obra = Costo * el valor del campo "O-D/KIT". - El sistema valida la Cantidad solo ingreso de numeros. - El sistema realiza calculos dinamicos es decir si en la Base de Datos cambia el Costo del Perfil, el T/KIT del detalle cambia automaticamente. - El sistema agrega el detalle en un Grid. - El sistema valida los campos obligatorios: Profesion, Perfil, Cantidad.		
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - Los Calculos se efectuan perfectamente. - La Mano de Obra se guarda exitosamente. - El Sistema emite todos los mensajes en esta prueba.		

**Figura 108. Prueba de aceptación agregar detalle mano de obra**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de mano de obra en el kit
NUMERO CASO DE PRUEBA:	CP12.02
NUMERO HISTORIA DE USUARIO:	HU12
NOMBRE CASO DE PRUEBA:	Editar detalle Mano de Obra
DESCRIPCION:	Se realiza el modificado de detalle de Materia de Obra con los Datos: Cantidad.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrada la Mano de Obra a editar. El usuario ejecuta el Caso de Prueba(Buscar KIT).
ENTRADAS:	1. El usuario da click en el boton Modificar del detalle de Mano de Obra. 2. El usuario realiza los cambios en los campos. 3. El usuario da click en el boton Modificar.
RESULTADO ESPERADO:	- El sistema calcula el campo O-D/KIT del detalle de la Mano de Obra = Cantidad * el valor del campo "Dias/Kit" del Kit . - El sistema calcula el campo T/KIT. del detalle de la Mano de Obra = Costo * el valor del campo "O-D/KIT". - El sistema valida la Cantidad solo ingreso de numeros. - El sistema realiza calculos dinamicos es decir si en la Base de Datos cambia el Costo del Perfil, el T/KIT del detalle cambia automaticamente. - El sistema modifica con los nuevos datos el detalle en el Grid. - El sistema valida los campos obligatorios: Cantidad.
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - Los Calculos se efectuan perfectamente. - El detalle de Mano de Obra se actualiza exitosamente. - El Sistema emite todos los mensajes en esta prueba.

**Figura 109. Prueba de aceptación editar detalle mano de obra**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Registro de mano de obra en el kit
NUMERO CASO DE PRUEBA:	CP12.03
NUMERO HISTORIA DE USUARIO:	HU12
NOMBRE CASO DE PRUEBA:	Eliminar detalle Mano de Obra
DESCRIPCION:	Se realiza el eliminado de detalle de Mano de Obra con los Datos: Profesion, Perfil, Costo, Cantidad, O-D/KIT, T/KIT.
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba(Agregar detalle Mano de Obra).
ENTRADAS:	1. El usuario da click en el boton eliminar.
RESULTADO ESPERADO:	- El sistema realiza un eliminado físico. - El sistema elimina el detalle del Grid.
EVALUACION:	- El detalle de Mano de Obra se elimina exitosamente. - El Sistema emite todos los mensajes en esta prueba.

**Figura 110. Prueba de aceptación eliminar detalle mano de obra**

### 5.1.3 Pruebas de aceptación tercera iteración.

Para este módulo se especifica las pruebas de aceptación correspondientes a las historias de usuario pertenecientes a gestión control de ofertas.

- Crear una oferta.
- Ingreso de estados a la oferta.
- Ingreso de contactos a la oferta.

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Crear una oferta		
NUMERO CASO DE PRUEBA:	CP20.01	NUMERO HISTORIA DE USUARIO:	HU20
NOMBRE CASO DE PRUEBA:	Guardar Oferta		
DESCRIPCION:	Se realiza la creacion de la Oferta con los datos: Codigo, Nombre, Linea de Negocio, Descripcion, estado.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. Tienen que estar registradas las Lineas de Negocio.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Ofertas.</li> <li>4. El usuario da click en el boton Nuevo.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema genera un codigo automatico de acuerdo al formato: OFER. + "LINEA DE NEGOCIO" + NUMERO SECUENCIAL DE TRES DIGITOS.</li> <li>- El sistema transforma el Nombre a mayusculas.</li> <li>- El sistema emite un mensaje de confirmacion de guardado.</li> <li>- El sistema valida los campos obligatorios: Nombre, Linea de Negocio.</li> <li>- El sistema valida que esté ingresado un estado inicial a la Oferta</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- La Oferta se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>		

**Figura 111. Prueba de aceptación guardar oferta**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Crear una oferta		
NUMERO CASO DE PRUEBA:	CP20.02	NUMERO HISTORIA DE USUARIO:	HU20
NOMBRE CASO DE PRUEBA:	Buscar Oferta		
DESCRIPCION:	Se realiza la busqueda de la Oferta por el Nombre.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la Oferta a buscar.		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Ofertas.</li> <li>3. El usuario ingresa el parametro a buscar.</li> <li>4. El usuario selecciona el registro buscado.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema despliega todas las coincidencias en una lista.</li> <li>- Los datos que se despliegan en la lista son: Codigo, Nombre, Descripcion.</li> <li>- El sistema al seleccionar la Oferta carga en la pantalla los datos de la misma.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- La Oferta se busca exitosamente.</li> <li>- Los datos en la lista se despliegan correctamente.</li> </ul>		

**Figura 112. Prueba de aceptación buscar oferta**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Crear una oferta
NUMERO CASO DE PRUEBA:	CP20.03
NUMERO HISTORIA DE USUARIO:	HU20
NOMBRE CASO DE PRUEBA:	Editar Oferta
DESCRIPCION:	Se realiza la edición de la Oferta con los datos: Nombre, Descripción.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la Oferta a editar. Se debe primero buscar la Oferta.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Ofertas. 3. El usuario busca el registro a editar. 4. El usuario hace los cambios en los campos. 5. El usuario da click en el boton Actualizar.
RESULTADO ESPERADO:	- El sistema transforma el Nombre a mayusculas. - El sistema emite un mensaje de confirmacion de actualizado. - El sistema valida los campos obligatorios: Nombre. - El sistema valida que esté ingresado un estado inicial a la Oferta
EVALUACION:	- La Oferta se actualiza exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 113. Prueba de aceptación editar oferta**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de estados a la oferta
NUMERO CASO DE PRUEBA:	CP21.01
NUMERO HISTORIA DE USUARIO:	HU21
NOMBRE CASO DE PRUEBA:	Agregar Estado a la Oferta
DESCRIPCION:	Se realiza el agregado de Estado con los Datos: Estado, Fecha de Estado, Observacion, Siguiete Estado, Fecha Siguiete Estado, Siguiete Gestion.
CONDICIONES DE EJECUCION:	Tienen que estar registrados los Estados de Proyecto. El usuario ejecuta el Caso de Prueba(Guardar Oferta).
ENTRADAS:	1. El usuario llena los campos en la pantalla. 2. El usuario da click en el boton Agregar Estado.
RESULTADO ESPERADO:	- Si el usuario escoje el estado Perdido en sistema agregara un campo de Empresa Ganadora de la Oferta, y ya no un siguiente Estado. - El sistema valida los campos obligatorios: Estado, Fecha de Estado, Observacion, Siguiete Estado, Fecha Siguiete Estado, Siguiete Gestion. - El sistema agrega el detalle de Estado en un Grid.
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - El Estado se agrega exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 114. Prueba de aceptación agregar estado a la oferta**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de estados a la oferta
NUMERO CASO DE PRUEBA:	CP21.02
NUMERO HISTORIA DE USUARIO:	HU21
NOMBRE CASO DE PRUEBA:	Eliminar Estado de la Oferta
DESCRIPCION:	Se realiza el eliminado de Estado con los Datos: Estado, Fecha de Estado, Observacion, Siguiente Estado, Fecha Siguiente Estado, Siguiente Gestion.
CONDICIONES DE EJECUCION:	Tienen que estar registrado el Estado de Oferta a Eliminar. El usuario ejecuta el Caso de Prueba (Buscar Oferta).
ENTRADAS:	1. El usuario da click en el boton eliminar. 2. El usuario confirma el eliminado.
RESULTADO ESPERADO:	- El sistema realiza un eliminado fisico. - El sistema elimina el detalle del Grid.
EVALUACION:	- El detalle de Estado de Oferta se elimina exitosamente. - El Sistema emite todos los mensajes de esta prueba.

Figura 115. Prueba de aceptación eliminar estado de la oferta

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de contactos a la oferta
NUMERO CASO DE PRUEBA:	CP22.01
NUMERO HISTORIA DE USUARIO:	HU22
NOMBRE CASO DE PRUEBA:	Agregar Contacto a la Oferta
DESCRIPCION:	Se realiza el agregado de un detalle de Contacto a la Oferta.
CONDICIONES DE EJECUCION:	Tienen que estar registrados los Contactos en la Base de Datos. El usuario ejecuta el Caso de Prueba(Guardar Oferta).
ENTRADAS:	1. El usuario busca por el Nombre el Contacto a Agregar. 2. El usuario da click en el Contacto buscado.
RESULTADO ESPERADO:	- Cada que el usuario escribe un caracter en el campo Nombre el sistema busca todas las coincidencias en la Base de Datos. - El sistema agrega el detalle del Contacto a un Grid con los Datos: N°, Nombre, Telefono1, Telefono2, Celular1, Celular2, Email. - Los datos son dinamicos es decir que si se modifican los Contactos automaticamente se cambiaran en la Oferta.
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - El Contacto se agrega exitosamente.

Figura 116. Prueba de aceptación agregar contacto a la oferta

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de contactos a la oferta
NUMERO CASO DE PRUEBA:	CP22.02
NUMERO HISTORIA DE USUARIO:	HU22
NOMBRE CASO DE PRUEBA:	Crear Contacto
DESCRIPCION:	Se realiza la creación de Contacto con los datos: Nombre, Telefono1, Telefono2, Celular1, Celular2, Email, Direccion, Estado.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema.
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Ofertas.</li> <li>4. El usuario da click en el boton Administrar Contactos.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el boton Guardar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema transforma el Nombre a mayusculas.</li> <li>- El sistema guarda con un Estado True.</li> <li>- El sistema emite un mensaje de confirmacion de creado.</li> <li>- El sistema valida los campos obligatorios: Nombre, Telefono1 y Celular1.</li> <li>- El sistema valida que los campos Telefono1, Telefono2, Celular1, Celular2 sean solo numeros.</li> <li>- El Email tienen que ser validado de acuerdo a la estructura del mismo.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El Contacto se crea exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

Figura 117. Prueba de aceptación crear contacto

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de contactos a la oferta
NUMERO CASO DE PRUEBA:	CP22.03
NUMERO HISTORIA DE USUARIO:	HU22
NOMBRE CASO DE PRUEBA:	Eliminar Contacto de Oferta
DESCRIPCION:	Se realiza el eliminado de un detalle de Contacto de la Oferta.
CONDICIONES DE EJECUCION:	Tienen que estar registrado el Estado de Oferta a Eliminar. El usuario ejecuta el Caso de Prueba (Buscar Oferta).
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario da click en el boton eliminar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema realiza un eliminado.</li> <li>- El sistema elimina el detalle del Grid.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- El detalle de Contacto de Oferta se elimina exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

Figura 118. Prueba de aceptación eliminar contacto de oferta

#### 5.1.4 Pruebas de aceptación cuarta iteración.

Para este módulo se especifica las pruebas de aceptación correspondientes a las historias de usuario pertenecientes a gestión de cotizaciones.

- Ingreso de datos a la cabecera de la cotización.
- Añadir kits a la cotización.
- Añadir otros gastos a la cotización.
- Añadir datos informativos a la cotización.



PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso datos a la cabecera de la cotizacion		
NUMERO CASO DE PRUEBA:	CP30.01	NUMERO HISTORIA DE USUARIO:	HU30
NOMBRE CASO DE PRUEBA:	Guardar Cotización		
DESCRIPCION:	Se realiza la creación de la Cotización con los datos: Oferta, Contacto, Descripción, Tiempo Ejecución, Código, Fecha Emisión, Días de Vigencia, Factor ganancia, Estado.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. Tienen que estar registrada en la Base de Datos la Oferta.		
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Cotización. 4. El usuario da click en el boton Nuevo. 5. El usuario llena los datos de la pantalla. 6. El usuario da click en el boton Guardar.		
RESULTADO ESPERADO:	- El boton nuevo limpia todos los datos de la pantalla. - El sistema genera un código automático basado en el formato "COTI." + SECUENCIAL DE CUATRO DIGITOS. - El sistema transforma la Descripción a mayúsculas. - El sistema valida que los campos Días de Vigencia, Tiempo de Ejecución, Factor Ganancia sean solo numeros. - Cuando se ingresa un valor en el campo Factor Ganancia el sistema realiza un reCalcular de costos en la Cotización. - El sistema emite un mensaje de confirmación de guardado. - El sistema valida los campos obligatorios: Oferta, Contacto, Tiempo Ejecución, Fecha Emisión, Días de Vigencia, Factor ganancia. - Las Ofertas que aparecen en la cotizacion seran las que están en estado de CIRO.		
EVALUACION:	- El boton nuevo limpia todos los campos. - Las validaciones se efectuan correctamente en esta prueba. - Los Calculars se efectuan perfectamente. - La Cotización se guarda exitosamente. - El Sistema emite todos los mensajes en esta prueba.		

Figura 119. Prueba de aceptación guardar cotización

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso datos a la cabecera de la cotizacion		
NUMERO CASO DE PRUEBA:	CP30.02	NUMERO HISTORIA DE USUARIO:	HU30
NOMBRE CASO DE PRUEBA:	Buscar Cotización		
DESCRIPCION:	Se realiza la búsqueda de la Cotización por: Nombre de Cotización, Código o Fecha de Emisión.		
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. En el sistema tiene que estar registrada la Cotización a buscar.		
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Cotización. 3. El usuario da click en el boton Buscar. 4. El usuario ingresa el parametro a buscar. 5. El usuario selecciona la Cotización buscada.		
RESULTADO ESPERADO:	- El sistema despliega todas las coincidencias en una lista. - Los datos que se desplegan en la lista son: N°, Codigo, Nombre Cotizacion, Fecha Emisión. - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Kit). - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Gasto Aumenta Cotización). - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Gasto Recalcula Cotización). - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Gasto Extras Cotización).		
EVALUACION:	- La Cotización se busca exitosamente. - Los datos en la lista se desplegan correctamente. - Se ejecuta efectivamente los Casos de Prueba.		

Figura 120. Prueba de aceptación buscar cotización

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso datos a la cabecera de la cotizacion		
NUMERO CASO DE PRUEBA:	CP30.03	NUMERO HISTORIA DE USUARIO:	HU30
NOMBRE CASO DE PRUEBA:	Editar Cotización		
DESCRIPCION:	Se realiza la edición de la Cotización con los datos: Nombre Cotización, Descripción, Tiempo Ejecución, Código, Fecha Emisión, Días de Vigencia, Factor ganancia, Estado.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Buscar Cotización).		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario hace los cambios en los campos.</li> <li>2. El usuario da click en el boton Actualizar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema transforma el Nombre de la Cotización y la Descripción a mayúsculas.</li> <li>- El sistema valida que los campos Dias de Vigencia, Tiempo de Ejecucion, Factor Ganancia sean solo numeros.</li> <li>- Cuando se ingresa un valor en el campo Factor Ganancia el sistema realiza un reCalcular de costos en la Cotización.</li> <li>- El sistema emite un mensaje de confirmación de actualizado.</li> <li>- El sistema valida los campos obligatorios: Oferta, Nombre de Cotización, Tiempo Ejecución, Fecha Emisión, Días de Vigencia, Factor ganancia.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- Los Calculars se efectuan perfectamente.</li> <li>- La Cotización se actualizó exitosamente.</li> <li>- El Sistema emite todos los mensajes en esta prueba.</li> </ul>		

**Figura 121. Prueba de aceptación editar cotización**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso datos a la cabecera de la cotizacion		
NUMERO CASO DE PRUEBA:	CP30.04	NUMERO HISTORIA DE USUARIO:	HU30
NOMBRE CASO DE PRUEBA:	Copiar Cotización		
DESCRIPCION:	Se realiza una copia total de toda la Cotización, para poder generar una nueva.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Buscar Cotización).		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario da click en el boton Copiar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema genera una copia de la Cotización a excepción del Código y setea en todos los campos de la pantalla.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- La Cotización se copia exitosamente.</li> <li>- Los datos en la pantalla se setean correctamente.</li> </ul>		

**Figura 122. Prueba de aceptación copiar cotización**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Calcular Cotización
NUMERO CASO DE PRUEBA:	CP30.05   NUMERO HISTORIA DE USUARIO: NA
NOMBRE CASO DE PRUEBA:	Calcular Cotización
DESCRIPCION:	Se realiza el Calcular de la Cotización, cada que el sistema lo pida.
CONDICIONES DE EJECUCION:	Se Calcula una Cotización cada que una Historia de Usuario de la Cuarta Iteración lo solicite.
ENTRADAS:	1. El usuario realiza un agregado, edición o eliminación de un Kit o un Gasto, o el usuario realiza un cambio en el Factor Ganancia de la Cotización.
RESULTADO ESPERADO:	<p>- El sistema muestra un Grid Lista Kits con los siguientes Datos:  Codigo = Codigo del Kit,  Cant = Por default es de 1,  Nombre = Nombre del Kit,  Unidad = Unidad del Kit,  Costo Real = Cant_Aum * Total del Kit,  Costo_Otros = Costo Real + ((Costo Real / SUMA(Costo Real)) * Total Gastos tipo Aumenta Cotización).  Precio_Total = Costo_Otros * Factor Ganancia de la Cotizacion %.  Precio_ReCalcular = Precio_Total - Total Gastos tipo Recalcula Cotizacion.  Precio_Unitario = Precio_ReCalcular / Cant_Aum</p> <p>- El sistema muestra un Grid Resumen con los siguientes Datos:  N° = Número secuencial,  Cant = Cant_Aum,  Unidad = Unidad del Kit,  Nombre = Nombre del Kit,  Precio Unitario = Precio_Unitario,  Total = Precio_ReCalcular</p> <p>- El sistema muestra un Grid Resumen Totales con los siguientes Datos:  Subtotal = Suma(Total del Grid Resumen ),  Iva= Subtotal * 12%,  Total = Subtotal + Iva</p>
EVALUACION:	<p>- Los Calculars se efectuan correctamente  - Los Grids se presentan efectivamente.</p>

Figura 123. Prueba de aceptación calcular cotización

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Añadir kits a la cotizacion
NUMERO CASO DE PRUEBA:	CP31.01   NUMERO HISTORIA DE USUARIO: HU31
NOMBRE CASO DE PRUEBA:	Agregar Kit
DESCRIPCION:	Se realiza el agregado de detalle de Kit con los Datos: N°, Codigo, Nombre, Unidad, Linea, Dias, Kit, Dias/Kit, Kit/Dias, Total Materia Prima, Total Mano de Obra, Total Costo Indirecto, Cantidad, Total Kit.
CONDICIONES DE EJECUCION:	Tienen que estar registrados los Kits en la Base de Datos. El usuario ejecuta el Caso de Prueba (Guardar Cotización).
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario busca el Kit a agregar.</li> <li>2. El usuario selecciona el Kit a agregar.</li> </ol>
RESULTADO ESPERADO:	<p>- Cada que el usuario ingresa un caracter en el campo Kit, el sistema automaticamente busca por Nombre o Codigo en la Base de Datos.  - El sistema agrega el detalle del Kit con todos los datos en un Grid.  - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Kit).</p>
EVALUACION:	<p>- Las busquedas del kit se efectuan correctamente.  - El Kit se agrega exitosamente.  - Se ejecuta efectivamente el Caso de Prueba (Calcular Kit).</p>

Figura 124. Prueba de aceptación agregar kit

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Añadir kits a la cotizacion
NUMERO CASO DE PRUEBA:	CP31.02   NUMERO HISTORIA DE USUARIO: HU31
NOMBRE CASO DE PRUEBA:	Consultar Kit
DESCRIPCION:	Se realiza la consulta de la Materia Prima y Mano de Obra del Kit.
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Kit).
ENTRADAS:	1. El usuario da click en el botón Consultar del Kit.
RESULTADO ESPERADO:	- El sistema carga un grid "Lista Materia Prima" con los datos de la Materia Prima del Kit: Nº, Codigo, Nombre, Unidad, Cantidad, Precio, Total. - El sistema carga un grid "Lista Mano de Obra" con los datos de la Mano de Obra del Kit: Nº, Profesión, Perfil, Costo Directo, Cantidad, O-D/KIT, T/KIT.
EVALUACION:	- El Kit se consulta correctamente. - Los datos en los Grids se despliegan correctamente.

Figura 125. Prueba de aceptación consultar kit

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Añadir kits a la cotizacion
NUMERO CASO DE PRUEBA:	CP31.03   NUMERO HISTORIA DE USUARIO: HU31
NOMBRE CASO DE PRUEBA:	Editar Kit
DESCRIPCION:	Se realiza el editado de un detalle de Kit de la Cotización con los datos(Nombre, Dias, Kits).
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Kit).
ENTRADAS:	1. El usuario da click en el botón Modificar. 2. El usuario edita los campos permitidos. 3. El usuario da click en el boton Modificar.
RESULTADO ESPERADO:	- El sistema ejecuta automaticamente el Caso de Prueba (Calcular Kit).
EVALUACION:	- El Kit se edita correctamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Kit).

Figura 126. Prueba de aceptación editar kit

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Añadir kits a la cotizacion
NUMERO CASO DE PRUEBA:	CP31.04   NUMERO HISTORIA DE USUARIO: HU31
NOMBRE CASO DE PRUEBA:	Eliminar Kit
DESCRIPCION:	Se realiza el eliminando de un detalle de Kit de la Cotización.
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Kit).
ENTRADAS:	1. El usuario da click en el botón Eliminar.
RESULTADO ESPERADO:	- El sistema elimina el detalle de Kit de la Cotización - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).
EVALUACION:	- El Kit se elimina correctamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).

Figura 127. Prueba de aceptación eliminar kit

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Calcular Kit
NUMERO CASO DE PRUEBA:	CP31.05
NUMERO HISTORIA DE USUARIO:	NA
NOMBRE CASO DE PRUEBA:	Calcular Kit
DESCRIPCION:	Se realiza el procedimiento interno Calcular Kit cada que el sistema lo pida.
CONDICIONES DE EJECUCION:	Se Calcula un Kit cada que una Historia de Usuario de la Cuarta Iteración lo solicite.
ENTRADAS:	1. El usuario realiza una edición de un detalle de Materia Prima o Mano de Obra del Kit.
RESULTADO ESPERADO:	<p>- El sistema realiza el Calcular de los siguientes campos:  Dias_Kits = Dias / Kits,  Kits_dias = Kits / Dias,  Total Detalle Materia Prima = Cantidad MP * Costo MP.  Total Materia Prima = Suma(Total Detalle Materia).</p> <p>O-D/KIT = Dias_Kits * Cantidad MO,  T/KIT = O-D/KIT * Costo MO.  Total Mano de Obra = Suma(T/KIT).</p> <p>Total Costo Indirecto = Suma (O-D/KIT) * Costo Directo del Sistema.  Total Kit = Total Costo Indirecto + Total Materia Prima + Total Mano de Obra.</p> <p>- El sistema setea los nuevos valores en los Grids  - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).</p>
EVALUACION:	<p>- Los Calculars se efectuan correctamente  - Los datos se setean efectivamete en el Grid.  - Se ejecuta efectivamete el Caso de Prueba (Calcular Cotización).</p>

**Figura 128. Prueba de aceptación calcular kit**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Añadir kits a la cotizacion
NUMERO CASO DE PRUEBA:	CP31.06
NUMERO HISTORIA DE USUARIO:	HU31
NOMBRE CASO DE PRUEBA:	Editar Materia Prima
DESCRIPCION:	Se realiza el editado de un detalle de Materia Prima de Kit de la Cotización con los datos(Nombre, Cantidad, Precio).
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Consultar Kit).
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario da click en el botón Modificar.</li> <li>2. El usuario edita los campos permitidos.</li> <li>3. El usuario da click en el boton Modificar.</li> </ol>
RESULTADO ESPERADO:	- El sistema ejecuta automaticamente el Caso de Prueba (Calcular Kit).
EVALUACION:	<p>- El detalle de Materia Prima se edita correctamente.  - Se ejecuta efectivamete el Caso de Prueba (Calcular Kit).</p>

**Figura 129. Prueba de aceptación editar materia prima**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Añadir kits a la cotizacion
<b>NUMERO CASO DE PRUEBA:</b>	CP31.07
<b>NUMERO HISTORIA DE USUARIO:</b>	HU31
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Mano de Obra
<b>DESCRIPCION:</b>	Se realiza el editado de un detalle de Mano de Obra de Kit de la Cotización con los datos (Perfil, Costo por Dia, Cantidad).
<b>CONDICIONES DE EJECUCION:</b>	El usuario ejecuta el Caso de Prueba (Consultar Kit).
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario da click en el botón Modificar.</li> <li>2. El usuario edita los campos permitidos.</li> <li>3. El usuario da click en el boton Modificar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	- El sistema ejecuta automaticamente el Caso de Prueba (Calcular Kit).
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- El detalle de Mano de Obra se edita correctamente.</li> <li>- Se ejecuta efectivamente el Caso de Prueba (Calcular Kit).</li> </ul>

**Figura 130. Prueba de aceptación editar mano de obra**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Añadir otros gastos a la cotizacion
<b>NUMERO CASO DE PRUEBA:</b>	CP32.01
<b>NUMERO HISTORIA DE USUARIO:</b>	HU32
<b>NOMBRE CASO DE PRUEBA:</b>	Agregar Gasto Aumenta Cotización
<b>DESCRIPCION:</b>	Se realiza el agregado de Gasto que Aumenta en la Cotizacion pero el valor se reparte para todos los detalles de los Kits de la Cotización, los Datos son: Unidad, Cantidad, Descripción, Valor.
<b>CONDICIONES DE EJECUCION:</b>	El usuario ejecuta el Caso de Prueba (Guardar Cotización).
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa los campos en la pantalla.</li> <li>2. El usuario da click en el boton Agregar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	<ul style="list-style-type: none"> <li>- El sistema valida los campos obligatorios: Unidad, Cantidad, Descripción, Valor.</li> <li>- El sistema valida que en los campos Cantidad y Valor solo se ingresen números.</li> <li>- El sistema agrega los datos en el Grid Lista Otros Aumenta Cotización.</li> <li>- El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).</li> </ul>
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El detalle de Gasto se agrega exitosamente.</li> <li>- Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).</li> </ul>

**Figura 131. Prueba de aceptación agregar gasto aumenta cotización**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir otros gastos a la cotización		
NUMERO CASO DE PRUEBA:	CP32.02	NUMERO HISTORIA DE USUARIO:	HU32
NOMBRE CASO DE PRUEBA:	Eliminar Gasto Aumenta Cotización		
DESCRIPCION:	Se realiza el eliminado de Gasto que Aumenta en la Cotización		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Gasto Aumenta Cotización).		
ENTRADAS:	1. El usuario da click en el boton Eliminar del detalle.		
RESULTADO ESPERADO:	- El sistema elimina los datos del Grid Lista Otros Aumenta Cotización. - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).		
EVALUACION:	- El detalle de Gasto se elimina exitosamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).		

**Figura 132. Prueba de aceptación eliminar gasto aumenta cotización**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir otros gastos a la cotización		
NUMERO CASO DE PRUEBA:	CP32.03	NUMERO HISTORIA DE USUARIO:	HU32
NOMBRE CASO DE PRUEBA:	Agregar Gasto Recalcula Cotización		
DESCRIPCION:	Se realiza el agregado de Gasto que Recalcula Cotización quitando un porcentaje a cada detalle de Kit y aumentando como un detalle adicional, los Datos son: Unidad, Cantidad Porcentaje, Descripción.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Guardar Cotización).		
ENTRADAS:	1. El usuario ingresa los campos en la pantalla. 2. El usuario da click en el boton Agregar.		
RESULTADO ESPERADO:	- El sistema valida los campos obligatorios: Unidad, Cantidad, Descripción. - El sistema valida que en el campo Cantidad solo se ingresen números. - El sistema agrega los datos en el Grid Lista Otros Recalcula Cotización. - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).		
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - El detalle de Gasto se agrega exitosamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).		

**Figura 133. Prueba de aceptación agregar gasto recalcula cotización**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir otros gastos a la cotización		
NUMERO CASO DE PRUEBA:	CP32.04	NUMERO HISTORIA DE USUARIO:	HU32
NOMBRE CASO DE PRUEBA:	Eliminar Gasto Recalcula Cotización		
DESCRIPCION:	Se realiza el eliminado de Gasto que Recalcula en la Cotización		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Gasto Recalcula Cotización).		
ENTRADAS:	1. El usuario da click en el boton Eliminar del detalle.		
RESULTADO ESPERADO:	- El sistema elimina los datos del Grid Lista Otros Recalcula Cotización. - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).		
EVALUACION:	- El detalle de Gasto se elimina exitosamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).		

**Figura 134. Prueba de aceptación eliminar gasto recalcula cotización**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir otros gastos a la cotización		
NUMERO CASO DE PRUEBA:	CP32.05	NUMERO HISTORIA DE USUARIO:	HU32
NOMBRE CASO DE PRUEBA:	Agregar Gasto Extras Cotización		
DESCRIPCION:	Se realiza el agregado de Gasto que agrega un Producto de la Base de Datos del ERP mas un porcentaje de ganancia a la Cotización, los Datos son: Producto, Unidad, Cantidad, Valor.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Guardar Cotización).		
ENTRADAS:	1. El usuario busca en la Base de Datos del ERP el Producto a agregar 2. El usuario da click en el producto a agregar. 3. El usuario ingresa la información en los campos. 4. El usuario da click en el boton Agregar.		
RESULTADO ESPERADO:	- Cada que el usuario ingresa un caracter en el campo Producto, el sistema despliega una lista con todas las coincidencias. - Al momento de seleccionar el Producto el sistema setea el costo del mismo y el valor = costo * factor ganancia. - El sistema valida los campos obligatorios: Producto, Unidad, Cantidad, Valor. - El sistema valida que en los campos Cantidad y Valor solo se ingresen números. - El sistema agrega los datos en el Grid Lista Otros Extras. - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).		
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - El detalle de Gasto se agrega exitosamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).		

**Figura 135. Prueba de aceptación agregar gasto extras cotización**



PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir otros gastos a la cotización		
NUMERO CASO DE PRUEBA:	CP32.06	NUMERO HISTORIA DE USUARIO:	HU32
NOMBRE CASO DE PRUEBA:	Eliminar Gasto Extras Cotización		
DESCRIPCION:	Se realiza el eliminado de Gasto Extra en la Cotización		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Gasto Extras Cotización).		
ENTRADAS:	1. El usuario da click en el boton Eliminar del detalle.		
RESULTADO ESPERADO:	- El sistema elimina los datos del Grid Lista Otros Extras. - El sistema ejecuta automaticamente el Caso de Prueba (Calcular Cotización).		
EVALUACION:	- El detalle de Gasto se elimina exitosamente. - Se ejecuta efectivamente el Caso de Prueba (Calcular Cotización).		

**Figura 136. Prueba de aceptación eliminar gasto extras cotización**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir datos informativos en la cotización		
NUMERO CASO DE PRUEBA:	CP33.01	NUMERO HISTORIA DE USUARIO:	HU33
NOMBRE CASO DE PRUEBA:	Agregar Dato Informativo		
DESCRIPCION:	Se realiza el agregado de Dato Informativo a la Cotización que puede ser de cuatro tipos : Incluye, No Incluye, Forma de Pago, Otros; el dato para cada tipo es: Detalle.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Guardar Cotización).		
ENTRADAS:	1. El usuario escoje el tipo de Dato Informativo. 2. El usuario ingresa la información en los campos. 3. El usuario da click en el boton Agregar.		
RESULTADO ESPERADO:	- El sistema valida los campos obligatorios: Descripción. - El sistema agrega los datos en el Grid Lista Datos Informativos.		
EVALUACION:	- El detalle de Dato Informativo se agrega exitosamente.		

**Figura 137. Prueba de aceptación agregar dato informativo**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Añadir datos informativos en la cotización		
NUMERO CASO DE PRUEBA:	CP33.02	NUMERO HISTORIA DE USUARIO:	HU33
NOMBRE CASO DE PRUEBA:	Eliminar Dato Informativo		
DESCRIPCION:	Se realiza el eliminado del detalle de Dato Informativo de la Cotización.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Dato Informativo).		
ENTRADAS:	1. El usuario da click en el boton Eliminar del detalle.		
RESULTADO ESPERADO:	- El sistema elimina los datos del Grid Lista Datos Informativos.		
EVALUACION:	- El detalle de Dato Informativo se elimina exitosamente.		

**Figura 138. Prueba de aceptación eliminar dato informativo**

### 5.1.5 Pruebas de aceptación quinta iteración.

Para este módulo se especifica las pruebas de aceptación correspondientes a las historias de usuario pertenecientes a planificación del proyecto.

- Ingreso de datos al proyecto.
- Ingreso de clientes al proyecto.
- Ingreso de otros datos al proyecto.
- Ingreso de tareas al proyecto.

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Ingreso de datos al proyecto
<b>NUMERO CASO DE PRUEBA:</b>	CP40.01   <b>NUMERO HISTORIA DE USUARIO:</b>   HU40
<b>NOMBRE CASO DE PRUEBA:</b>	Guardar Proyecto
<b>DESCRIPCION:</b>	Se realiza la creación del Proyecto con los datos: Contrato, Nombre, Dirección, Teléfonos Obra, Código, Fecha, Forma de Pago.
<b>CONDICIONES DE EJECUCION:</b>	En el sistema tiene que estar registrado el usuario para logearse al sistema. Tiene que estar registrada la Cotización en la Base de Datos con estado Aceptada.
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario se logea en el sistema.</li> <li>2. El usuario da click en el boton Proyecto</li> <li>3. El usuario da click en el boton Nuevo.</li> <li>4. El usuario agrega la o los Contratos.</li> <li>5. El usuario llena los datos de la pantalla.</li> <li>6. El usuario da click en el botón Guardar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los datos de la pantalla.</li> <li>- El sistema genera un código automático basado en el formato "PLANI."+"SECUENCIAL DE CUATRO DIGITOS.</li> <li>- El sistema transforma los campos Nombre, Dirección y Forma de Pago a mayúsculas.</li> <li>- El sistema emite un mensaje de confirmación de guardado.</li> <li>- El sistema valida los campos obligatorios: Contrato, Nombre, Dirección, Telefonos Obra, Fecha, Forma de Pago.</li> </ul>
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- El boton nuevo limpia todos los campos.</li> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- El Proyecto se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

**Figura 139. Prueba de aceptación guardar proyecto**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de datos al proyecto
NUMERO CASO DE PRUEBA:	CP40.02   NUMERO HISTORIA DE USUARIO:   HU40
NOMBRE CASO DE PRUEBA:	Buscar Proyecto
DESCRIPCION:	Se realiza la búsqueda del Proyecto por los campos : Contrato, Nombre, Código.
CONDICIONES DE EJECUCION:	En el sistema tiene que estar registrado el usuario para logearse al sistema. Tiene que estar registrado el Proyecto en la Base de Datos.
ENTRADAS:	1. El usuario se logea en el sistema. 2. El usuario da click en el boton Proyecto 3. El usuario da click en el boton Buscar. 4. El usuario ingresa los datos de la búsqueda. 5. El usuario da click en el Proyecto.
RESULTADO ESPERADO:	- Cada que el usuario ingresa un caracter en el campo de búsqueda el sistema busca por Contrato, Nombre o Código. - El sistema despliega todas las coincidencias en una lista con los datos: N°, Código, Contrato, Nombre, Fecha. - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Kit). - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Gasto Aumenta Cotización). - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Gasto Recalcula Cotización). - El sistema ejecuta automaticamente el Caso de Prueba (Agregar Gasto Extras Cotización).
EVALUACION:	- El Proyecto se busca exitosamente. - Los datos en la lista se despliegan correctamente. - Se ejecuta efectivamente los Casos de Prueba.

**Figura 140. Prueba de aceptación buscar proyecto**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de datos al proyecto
NUMERO CASO DE PRUEBA:	CP40.03   NUMERO HISTORIA DE USUARIO:   HU40
NOMBRE CASO DE PRUEBA:	Editar Proyecto
DESCRIPCION:	Se realiza la modificación del Proyecto con los datos: Contrato, Nombre, Dirección, Teléfonos Obra, Fecha, Forma de Pago.
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Buscar Cotización).
ENTRADAS:	1. El usuario hace los cambios en los campos. 2. El usuario da click en el boton Actualizar.
RESULTADO ESPERADO:	- El sistema transforma los campos Nombre, Dirección y Forma de Pago a mayúsculas. - El sistema emite un mensaje de confirmación de actualizado. - El sistema valida los campos obligatorios: Contrato, Nombre, Dirección, Telefonos Obra, Fecha, Forma de Pago.
EVALUACION:	- Las validaciones se efectuan correctamente en esta prueba. - El Proyecto se actualiza exitosamente. - El Sistema emite todos los mensajes de esta prueba.

**Figura 141. Prueba de aceptación editar proyecto**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de personas involucradas al proyecto		
NUMERO CASO DE PRUEBA:	CP41.01	NUMERO HISTORIA DE USUARIO:	HU41
NOMBRE CASO DE PRUEBA:	Agregar Personas Involucradas		
DESCRIPCION:	Se realiza el agregado de Persona Involucrada con los Datos: N°, Tipo, Observación.		
CONDICIONES DE EJECUCION:	Tiene que estar registrado el contacto en la Base de Datos. El usuario ejecuta el Caso de Prueba (Guardar Proyecto).		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario ingresa el Nombre del Contacto a agregar</li> <li>2. El usuario selecciona el Contacto.</li> <li>3. El usuario ingresa los datos de la pantalla.</li> <li>4. El usuario da click en el botón Agregar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema despliega en un combo box los tipos de Personas Involucradas que son : Residente de Obra, Encargado de la Negociación, Gerente del Proyecto, Fiscalizador.</li> <li>- Cada que el usuario ingresa un caracter en el campo Nombre, el sistema busca el contacto a ingresar.</li> <li>- El sistema agrega el contacto en la Lista Personal Involucrado con los siguientes datos: Tipo, Nombre, Telefono1, Telefono2, Celular1, Celular2, Email, Dirección, Observación.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- Las búsquedas se efectuan correctamente.</li> <li>- La Persona Involucrada se agrega exitosamente.</li> </ul>		

**Figura 142. Prueba de aceptación agregar personas involucradas**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de personas involucradas al proyecto		
NUMERO CASO DE PRUEBA:	CP41.02	NUMERO HISTORIA DE USUARIO:	HU41
NOMBRE CASO DE PRUEBA:	Eliminar Personas Involucradas		
DESCRIPCION:	Se realiza el eliminado de Persona Involucrada.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Persona Involucrada).		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario da click en el botón Eliminar del detalle de Persona Involucrada.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema elimina el detalle de La Lista Personal Involucrado del Proyecto.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El detalle de Persona Involucrada se elimina correctamente.</li> </ul>		

**Figura 143. Prueba de aceptación eliminar personas involucradas**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de clientes al proyecto
NUMERO CASO DE PRUEBA:	CP42.01
NUMERO HISTORIA DE USUARIO:	HU42
NOMBRE CASO DE PRUEBA:	Agregar Cliente
DESCRIPCION:	Se realiza el agregado de Cliente con los Datos: N°, Tipo, Cliente, Observación.
CONDICIONES DE EJECUCION:	Tiene que estar registrado el cliente en la Base de Datos del ERP. El usuario ejecuta el Caso de Prueba (Guardar Proyecto).
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario ingresa el Nombre del Cliente a agregar</li> <li>2. El usuario selecciona el Cliente.</li> <li>3. El usuario ingresa los datos de la pantalla.</li> <li>4. El usuario da click en el botón Agregar.</li> </ol>
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema despliega en un combo box los tipos de Clientes que son : Contratante, Facturar.</li> <li>- Cada que el usuario ingresa un caracter en el campo Cliente, el sistema busca en la Base de Datos del ERP para el ingreso.</li> <li>- El sistema agrega el Cliente en la Lista de Clientes con los siguientes datos: N°, Tipo, Nombre, Ruc, Dirección, Teléfono, Observación.</li> </ul>
EVALUACION:	<ul style="list-style-type: none"> <li>- Las búsquedas se efectuan correctamente.</li> <li>- El Cliente se agrega exitosamente.</li> </ul>

**Figura 144. Prueba de aceptación agregar cliente**

PRUEBAS DE ACEPTACION	
CASO DE PRUEBA:	Ingreso de clientes al proyecto
NUMERO CASO DE PRUEBA:	CP42.02
NUMERO HISTORIA DE USUARIO:	HU42
NOMBRE CASO DE PRUEBA:	Eliminar Cliente
DESCRIPCION:	Se realiza el eliminado del Cliente.
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Cliente).
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario da click en el botón Eliminar del detalle de Cliente.</li> </ol>
RESULTADO ESPERADO:	- El sistema elimina el detalle de La Lista de Clientes del Proyecto.
EVALUACION:	- El detalle de Cliente se elimina correctamente.

**Figura 145. Prueba de aceptación eliminar cliente**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de otros datos al proyecto
<b>NUMERO CASO DE PRUEBA:</b>	CP43.01
<b>NUMERO HISTORIA DE USUARIO:</b>	HU43
<b>NOMBRE CASO DE PRUEBA:</b>	Agregar Otro Dato
<b>DESCRIPCION:</b>	Se realiza el agregado de Otro Dato con los Campos: N°, Tipo, Contacto, Observación.
<b>CONDICIONES DE EJECUCION:</b>	Tiene que estar registrado el contacto en la Base de Datos. El usuario ejecuta el Caso de Prueba (Guardar Proyecto).
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario ingresa el Nombre del Contacto a agregar</li> <li>2. El usuario selecciona el Contacto.</li> <li>3. El usuario ingresa los datos de la pantalla.</li> <li>4. El usuario da click en el botón Agregar.</li> </ol>
<b>RESULTADO ESPERADO:</b>	<ul style="list-style-type: none"> <li>- El sistema despliega en un combo box los tipos de Otros Datos que son : Pagos, Recepción Facturas.</li> <li>- Cada que el usuario ingresa un caracter en el campo Contacto, el sistema busca en la Base de Datos el Contacto para el ingreso.</li> <li>- El sistema agrega el Contacto en la "Lista Otros Datos" con los siguientes datos: N°, Tipo, Nombre, Telefono1, Telefono2, Celular1, Celular2, Email, Dirección, Observación.</li> </ul>
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- Las búsquedas se efectuan correctamente.</li> <li>- El Contacto se agrega exitosamente.</li> </ul>

**Figura 146. Prueba de aceptación agregar otro dato**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Registro de otros datos al proyecto
<b>NUMERO CASO DE PRUEBA:</b>	CP43.02
<b>NUMERO HISTORIA DE USUARIO:</b>	HU43
<b>NOMBRE CASO DE PRUEBA:</b>	Eliminar Otro Dato
<b>DESCRIPCION:</b>	Se realiza el eliminado del detalle Otro Dato.
<b>CONDICIONES DE EJECUCION:</b>	El usuario ejecuta el Caso de Prueba (Agregar Otro Dato).
<b>ENTRADAS:</b>	<ol style="list-style-type: none"> <li>1. El usuario da click en el botón Eliminar del detalle de Otro Dato.</li> </ol>
<b>RESULTADO ESPERADO:</b>	<ul style="list-style-type: none"> <li>- El sistema elimina el detalle de la "Lista Otros Datos" del Proyecto.</li> </ul>
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- El detalle de Cliente se elimina correctamente.</li> </ul>

**Figura 147. Prueba de aceptación eliminar otro dato**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Ingreso de tareas al proyecto
<b>NUMERO CASO DE PRUEBA:</b>	CP44.01
<b>NUMERO HISTORIA DE USUARIO:</b>	HU44
<b>NOMBRE CASO DE PRUEBA:</b>	Guardar Tarea
<b>DESCRIPCION:</b>	Se realiza el agregado de Tarea al Proyecto con los Datos: Nombre, Fecha Inicio, Fecha Finalización, Tarea Antecesora, Editar Nota.
<b>CONDICIONES DE EJECUCION:</b>	El usuario ejecuta el Caso de Prueba (Guardar Proyecto).
<b>ENTRADAS:</b>	1. El usuario ingresa los datos de la pantalla. 2. El usuario da click en el botón Guardar.
<b>RESULTADO ESPERADO:</b>	<ul style="list-style-type: none"> <li>- El sistema transforma el campo Nombre a mayúsculas.</li> <li>- El sistema valida que para ingresar una tarea primero se guarde la planificación.</li> <li>- El sistema valida que la Fecha Inicio sea igual o menor a la Fecha Finalización.</li> <li>- El sistema muestra en el campo Tarea Antecesora todas las tareas que tenga el Proyecto.</li> <li>- El sistema muestra un grafico con las Tareas Asignadas, Tiempo, Fecha Inicio, Fecha Fin.</li> <li>- El sistema brinda la posibilidad de visualizar las Tareas por Dia, Semana, o Mes.</li> </ul>
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- La Tarea se guarda exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

**Figura 148. Prueba de aceptación guardar tarea**

<b>PRUEBAS DE ACEPTACION</b>	
<b>CASO DE PRUEBA:</b>	Ingreso de tareas al proyecto
<b>NUMERO CASO DE PRUEBA:</b>	CP44.02
<b>NUMERO HISTORIA DE USUARIO:</b>	HU44
<b>NOMBRE CASO DE PRUEBA:</b>	Editar Tarea
<b>DESCRIPCION:</b>	Se realiza el editado de Tarea con los Datos: Nombre, Fecha Inicio, Fecha Finalización, Tarea Antecesora, Editar Nota.
<b>CONDICIONES DE EJECUCION:</b>	El usuario ejecuta el Caso de Prueba (Guardar Tarea).
<b>ENTRADAS:</b>	1. El usuario ingresa los datos de la pantalla. 2. El usuario da click en el botón Actualizar.
<b>RESULTADO ESPERADO:</b>	<ul style="list-style-type: none"> <li>- El sistema transforma el campo Nombre a mayúsculas.</li> <li>- El sistema valida que la Fecha Inicio sea igual o menor a la Fecha Finalización.</li> <li>- El sistema muestra en el campo Tarea Antecesora todas las tareas que tenga el Proyecto.</li> <li>- El sistema muestra un grafico con las Tareas Asignadas, Tiempo, Fecha Inicio, Fecha Fin.</li> <li>- El sistema brinda la posibilidad de visualizar las Tareas por Dia, Semana, o Mes.</li> </ul>
<b>EVALUACION:</b>	<ul style="list-style-type: none"> <li>- Las validaciones se efectuan correctamente en esta prueba.</li> <li>- La Tarea se actualiza exitosamente.</li> <li>- El Sistema emite todos los mensajes de esta prueba.</li> </ul>

**Figura 149. Prueba de aceptación editar tarea**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de tareas al proyecto		
NUMERO CASO DE PRUEBA:	CP44.03	NUMERO HISTORIA DE USUARIO:	HU44
NOMBRE CASO DE PRUEBA:	Eliminar Tarea		
DESCRIPCION:	Se realiza el eliminado de Tarea del Proyecto.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Guardar Tarea).		
ENTRADAS:	1. El usuario da click en la Tarea a eliminar. 2. El usuario da click en el botón Eliminar.		
RESULTADO ESPERADO:	- El sistema elimina la Tarea y elimina la tarea antecesora de todas las que dependan de la misma.		
EVALUACION:	- La Tarea se elimina exitosamente.		

Figura 150. Prueba de aceptación eliminar tarea

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de tareas al proyecto		
NUMERO CASO DE PRUEBA:	CP44.04	NUMERO HISTORIA DE USUARIO:	HU44
NOMBRE CASO DE PRUEBA:	Agregar Kit a la Tarea		
DESCRIPCION:	Se realiza el agregado de Kit del Contrato, a la Tarea del Proyecto con el dato: Cantidad.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Guardar Tarea).		
ENTRADAS:	1. El usuario selecciona el Kit a agregar. 2. El usuario ingresa los datos de la pantalla. 3. El usuario da click en el boton Agregar.		
RESULTADO ESPERADO:	- Al momento de seleccionar el Kit, el sistema carga en el campo Cantidad Disponible el saldo = Total Kit del Contrato - SUMA(Cantidad de Kit de todas las Tareas). - El sistema agrega en el Grid "Lista de Kits" el detalle del Kit con los datos: N°, Nombre, Cantidad. - El sistema valida que para agregar un kit a la tarea primero se debe guardar la tarea		
EVALUACION:	- El Kit se agrega exitosamente. - Los calculos se efectuan correctamente.		

Figura 151. Prueba de aceptación agregar kit a la tarea

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de tareas al proyecto		
NUMERO CASO DE PRUEBA:	CP44.05	NUMERO HISTORIA DE USUARIO:	HU44
NOMBRE CASO DE PRUEBA:	Eliminar Kit de Tarea		
DESCRIPCION:	Se realiza el eliminado de Kit de la Tarea del Proyecto.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Kit a la Tarea).		
ENTRADAS:	1. El usuario da click en el boton Eliminar del detalle de Kit a eliminar.		
RESULTADO ESPERADO:	- El sistema elimina del Grid "Lista de Kits" el detalle del Kit.		
EVALUACION:	- El detalle de Kit se elimina exitosamente.		

Figura 152. Prueba de aceptación eliminar kit de tarea



PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de tareas al proyecto		
NUMERO CASO DE PRUEBA:	CP44.06	NUMERO HISTORIA DE USUARIO:	HU44
NOMBRE CASO DE PRUEBA:	Agregar Responsable a la Tarea		
DESCRIPCION:	Se realiza el agregado de Responsable de la Tarea del Proyecto, con el dato: Tipo Responsable, Observación.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Guardar Tarea).		
ENTRADAS:	<ol style="list-style-type: none"> <li>1. El usuario selecciona el tipo de Responsable.</li> <li>2. El usuario ingresa los datos de la pantalla.</li> <li>3. El usuario da click en el boton Agregar.</li> </ol>		
RESULTADO ESPERADO:	<ul style="list-style-type: none"> <li>- El sistema transforma el campo Nombre a mayúsculas.</li> <li>- El sistema valida el campo obligatorio: Nombre.</li> <li>- Los tipos de Responsables son : Interno o Externo.</li> <li>- Si el tipo de Responsable seleccionado es Interno, el sistema cada que el usuario ingrese un caracter en el campo Nombre, busca en la Base de Datos del ERP los empleados que se encuentren ingresados en el mismo.</li> <li>- Si el tipo de Responsable seleccionado es Externo, el sistema deja ingresar libremente un Nombre.</li> <li>- El sistema agrega en el Grid "Lista de Responsables" el detalle del Responsable con los datos: N°, Tipo, Nombre, Observación.</li> </ul>		
EVALUACION:	<ul style="list-style-type: none"> <li>- El Responsable se agrega exitosamente.</li> <li>- Las búsquedas se efectuan correctamente.</li> <li>- Las validaciones se realizan exitosamente en esta prueba.</li> </ul>		

**Figura 153. Prueba de aceptación agregar responsable a la tarea**

PRUEBAS DE ACEPTACION			
CASO DE PRUEBA:	Ingreso de tareas al proyecto		
NUMERO CASO DE PRUEBA:	CP44.07	NUMERO HISTORIA DE USUARIO:	HU44
NOMBRE CASO DE PRUEBA:	Eliminar Responsable de Tarea		
DESCRIPCION:	Se realiza el eliminado de Responsable de la Tarea del Proyecto.		
CONDICIONES DE EJECUCION:	El usuario ejecuta el Caso de Prueba (Agregar Responsable a la Tarea).		
ENTRADAS:	1. El usuario da click en el boton Eliminar del detalle de Responsable a eliminar.		
RESULTADO ESPERADO:	- El sistema elimina del Grid "Lista de Responsables" el detalle del Responsable.		
EVALUACION:	- El detalle de Responsable se elimina exitosamente.		

**Figura 154. Prueba de aceptación eliminar responsable de tarea**

## 6 CONCLUSIONES Y RECOMENDACIONES

### 6.1 Conclusiones

- La aplicación no hubiera tenido una culminación exitosa si no se hubiera tenido un marco teórico lo suficientemente sólido acerca de la información del proceso, así como fuertes conocimientos técnicos y la suficiente infraestructura para llegar a implantar la solución.
- Los problemas presentados en las fases iniciales del proyecto concernientes a la metodología, se fueron depurando durante el transcurso de desarrollo de la aplicación, permitiendo finalmente conocer la simplicidad y flexibilidad que posee Extreme Programming como metodología ágil de desarrollo y mostrando que su aplicación involucra resultados visibles y funcionales a corto plazo.
- Se determinaron los problemas más antiguos en la planta así como se otorgó una herramienta a los líderes de cada área para que controlen los problemas generados.
- Durante el desarrollo de la aplicación, se observó que la utilización de PHP como lenguaje de programación y MySQL como motor de base de datos, constituyen herramientas adecuadas para el desarrollo de aplicaciones basadas en la metodología ágil XP, ya que ayudan a gestionar los cambios que pueden presentarse durante el implementación del sistema.
- A través de la presente tesis se han logrado abaratar costos y optimizar tiempos para obtener los indicadores de calidad así como facilitar la búsqueda de potenciales cuellos de botella y realizar la detección de problemas con lenta gestión más rápidamente.
- La planificación XP para la implementación de iteraciones, se basa en un entorno que involucra lo posible y lo deseable, en donde el tiempo ideal empleado para el desarrollo de un proyecto es alcanzar el tiempo estimado o

sobrepasarlo con un mínimo de diferencia, buscando de esta manera reducir el tiempo, la inversión y el riesgo que se presente en el periodo de desarrollo de la aplicación.

- En el seguimiento de iteraciones, es fundamental la comunicación entre el cliente y el desarrollador, ya que al ser manejado de forma paralela con la ejecución de iteraciones permite implementar las historias de usuario correspondientes a cada iteración y controlar las tareas necesarias para su desarrollo en base a tarjetas CRC; ayudando de esta manera a solucionar los posibles problemas que puedan presentarse de manera ágil y oportuna.
- Para que se pueda aplicar la metodología ágil se debe poseer experiencia trabajando con metodologías tradicionales, ya que la experiencia es la que predomina en los momentos cruciales del proyecto, además que el personal debe ser altamente motivado y con gran innovación.

## **6.2 Recomendaciones**

- Utilizar una Metodología Ágil puede ser una buena forma de empezar con el desarrollo de una aplicación sometida a constantes cambios, equipos pequeños de trabajo, mayor visibilidad, funcionalidad y menor riesgo.
- Debe hacerse lo posible por no realizar modificaciones a XP demasiado drásticas ya que se corre el riesgo de alterar la esencia de la metodología.
- Se deben fijar una serie de reglas generales en la comunicación con el cliente ya que por el grado de informalidad que la metodología presenta, pueden surgir diferencias que pongan en peligro la culminación exitosa del proyecto.
- Debe hacerse una capacitación al cliente sobre XP antes de iniciar el proyecto debido que este hace parte del equipo de desarrollo.

- Realizar las pruebas del sistema de forma continua y oportuna de manera que ayuden a reducir los posibles problemas que pueden presentarse a lo largo de la implementación de una iteración para lograr un producto funcional y de calidad.
- Mantener la comunicación entre usuarios y desarrolladores durante el proceso de implementación de la aplicación, con la finalidad de lograr una retroalimentación concreta y frecuente que permita desplegar resultados funcionales que cumplan con las expectativas del usuario.

## BIBLIOGRAFIA

- Anca, W., & Borda, W. (28 de Agosto de 2014). *Metodología XP*. Recuperado el 10 de Enero de 2015, de Procesos de software: <http://procesosdesoftware.wikispaces.com/METODOLOGIA+XP>
- Canedo, K. (04 de Febrero de 2015). *Netbeans*. Obtenido de Prezi: <https://prezi.com/-sauahxrt8cd/copy-of-netbeans/>
- Castro, L. (Febrero de 2012). *Que es HTML*. Obtenido de Givology: <http://aprenderinternet.about.com/od/Glosario/g/Que-Es-Html.htm>
- Cordero, M. (22 de Marzo de 2014). *Ventajas Apache*. Obtenido de Prezi: <https://prezi.com/br7auyci5leu/srevidore-web/>
- Delgado, E. (12 de Agosto de 2008). *Metodologías de desarrollo de software. ¿Cuál es el camino?* Recuperado el 10 de Enero de 2015, de Monografias.com: <http://www.monografias.com/trabajos60/metodologias-desarrollo-software/metodologias-desarrollo-software2.shtml>
- Garcia, K. (19 de Abril de 2013). *Apache*. Obtenido de Prezi: <https://prezi.com/2ygi9bjbbdy2/untitled-prezi/>
- González, E. (2006). *Un potente lenguaje de programación para crear paginas web*. Recuperado el 10 de Enero de 2015, de aprenderaprogramar.com: [http://www.aprenderaprogramar.com/index.php?option=com\\_content&id=492:i-que-es-php-y-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&Itemid=193](http://www.aprenderaprogramar.com/index.php?option=com_content&id=492:i-que-es-php-y-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&Itemid=193)
- Guerra, J. (19 de Enero de 2014). *Javascript*. Obtenido de Prezi: <https://prezi.com/hkmzd4jqhtwp/javascript/>

Jefry, G. (10 de Enero de 2015). *Desarrollo taller PHP*. Obtenido de Jimbo.com:  
<http://proyectoakro.jimdo.com/sistemas/>

Lamarca, M. (Octubre de 2012). *Hojas de estilo*. Obtenido de Hipertexto.com:  
<http://www.hipertexto.info/documentos/css.htm>

Lunarwar. (30 de Diciembre de 2014). *Teoría PHP. Mensaje en un Blog*. Obtenido de  
<http://lunartecnologia.blogspot.com/2014/12/que-es-php-para-que-sirve-php.html>

Madecyta. (2012). *Fundamentos del lenguaje PHP*. Recuperado el 10 de Enero de 2015, de PHP: <http://madecyta.blogspot.com/2012/05/fundamentos-del-lenguaje-de-php.html>

Marquetti Asociados. (2010). *El paradigma de la programación XP*. Recuperado el 10 de Enero de 2015, de Marquetti Asociados: <http://www.marquetti-asociados.com.ar/paradigma.php>

Paz, B. (s.f.). *6 buenos motivos para trabajar con PHP*. Recuperado el 10 de Enero de 2015, de LanceTalent: <http://www.lancetalent.com/blog/6-buenos-motivos-para-trabajar-con-php/>

PMOinformatica. (26 de Noviembre de 2012). *Los 5 valores de la programación extrema*. Recuperado el 10 de Enero de 2015, de PMOinformatica.com:  
<http://www.pmoinformatica.com/2012/11/los-5-valores-de-la-programacion.html>

Rana Negra. (2012). *Programación en PHP*. Recuperado el 10 de Enero de 2015, de Desarrollo Web: <http://www.rananegra.es/servicio/programacion-web-php>

Recargate, R. (28 de Julio de 2014). *MySQL*. Obtenido de Prezi:  
<https://prezi.com/al0xikxnqxuw/mysql-es-un-sistema-de-gestion-de-bases-de-datos-relacional/>

Rouse, M. (Enero de 2015). *MySQL*. Obtenido de TechTarget:  
<http://searchdatacenter.techtarget.com/es/definicion/MySQL>

Sanchez, G. (3 de Febrero de 2015). *Javascript y Java*. Obtenido de Prezi:  
<https://prezi.com/diewneqwnqvi/repaso-2/>


Universidad Unión Bolivariana. (2010). *Programación extrema*. Recuperado el 10 de Enero de 2015, de Ingeniería de Software:  
[http://ingenieriadesoftware.mex.tl/52753\\_XP---Extreme-Programing.html](http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html)

Valverde, D. (06 de Septiembre de 2007). *Introducción a la programación extrema*. Recuperado el 10 de Enero de 2015, de Software Developer:  
<http://www.davidvalverde.com/blog/introduccion-a-la-programacion-extrema-xp/?replytocom=6834>

Wells, D. (2009). *The Values of Extreme Programming*. Recuperado el 10 de Enero de 2015, de Extreme Programming:  
<http://www.extremeprogramming.org/values.html>

Wikipedia. (25 de Febrero de 2015). *AJAX*. Obtenido de AJAX:  
<http://es.wikipedia.org/wiki/AJAX>

## HOJA DE VIDA

	<p><b>Andrés Gabriel Peralta Tomalo</b></p> <p>Tungurahua OE2-82 y Villonaco , Quito          Fecha de nacimiento: 18 de septiembre de 1988          : 022660469 - 0987430299          : gapatmej@gmail.com          Cédula: 1720238706</p>
---	---

### **Aptitudes especiales**

- Honesto y leal con el grupo al que pertenezco
- Don de palabra y espíritu de liderazgo

### **Formación académica**

**1994 - 2000      Escuela**  
Escuela “Edmundo de Amicis”

**2000 – 2006      Colegio**  
“Instituto Nacional Mejía”  
Especialidad Químico Biólogo

**2007 - 2011 Universidad**  
Escuela Superior Politécnica del Ejército (ESPE)  
Tecnólogo en Sistemas de Información.  
Egresado en Enero 2011

### **Formación extra académica**

**2011              Suficiencia**  
Suficiencia en el idioma Inglés  
Escuela Superior Politécnica del Ejército (ESPE Av. 6 de Diciembre y Río Coca).

**2014              Curso**  
JAVA ADVANCED (Centro de Educación Continua).

### **Experiencia profesional**

**2015              Trabajo actual**  
Empresa: Tata Consultancy Services (TCS)  
(Joaquín Mancheno y Galo Plaza Lasso esquina)



Área: Desarrollo de Software  
Cargo: Analista IT

**2014 Desarrollador de Software**

Empresa: SISCONTI (La Kennedy)

Cargo: *Desarrollador de Software*

Función: Desarrollador de software en lenguaje PHP con base de datos MySQL.

### Idiomas

Inglés: Suficiencia.

### Informática

- **Office:** Alto Nivel
- **Bases de datos:** Nivel medio
- **Diseño Gráfico:** Nivel Medio
- **Desarrollo:** Nivel Medio
- **Sistemas Operativos:** Alto Nivel
- **Redes:** Alto Nivel

### Lenguajes de Programación

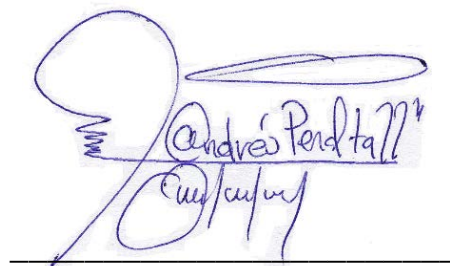
- **Visual Basic**
- **C#**
- **Java**
- **Jsp**
- **C++**
- **ASP**
- **HTML**
- **PHP.**

### Bases de Datos:

- **Oracle**
- **SQL Server**
- **Mysql**
- **Informix**

**HOJA DE LEGALIZACIÓN DE FIRMAS**

**ELABORADO POR**



Handwritten signature in blue ink, reading "Andrés Peralta" with a stylized flourish above the name. The signature is written over a horizontal line.

**PERALTA TOMALO ANDRÉS GABRIEL**

**DIRECTOR DE LA CARRERA**



Handwritten signature in blue ink, reading "Fernando Galarraga" with a stylized flourish above the name. The signature is written over a horizontal line.

**ING. FERNANDO GALARRAGA**