

**“DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA LA  
GESTIÓN DE LA INFORMACIÓN Y SEGUIMIENTO DE LOS TRABAJOS DE  
GRADO DE LA FACULTAD DE INGENIERÍA EN LA CORPORACIÓN  
UNIVERSITARIA AUTÓNOMA DEL CAUCA”**



CORPORACION UNIVERSITARIA  
**AUTÓNOMA**  
DEL CAUCA

**OCTAVIO ANDRÉS CIFUENTES DORADO**

**ANDRÉS FELIPE VÁSQUEZ**

**CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA**

**FACULTAD DE INGENIERÍA**

**INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POPAYÁN, NOVIEMBRE DE 2017**

**“DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA LA  
GESTIÓN DE LA INFORMACIÓN Y SEGUIMIENTO DE LOS TRABAJOS DE  
GRADO DE LA FACULTAD DE INGENIERÍA EN LA CORPORACIÓN  
UNIVERSITARIA AUTÓNOMA DEL CAUCA”**



**CORPORACION UNIVERSITARIA  
AUTONOMA  
DEL CAUCA**

**OCTAVIO ANDRÉS CIFUENTES DORADO**

**ANDRÉS FELIPE VÁSQUEZ**

**Trabajo de Grado para optar al título de Ingenieros de Sistemas  
Informáticos**

**Director**

**Esp. Julián Darío Bermúdez Trujillo**

**CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA**

**FACULTAD DE INGENIERÍA**

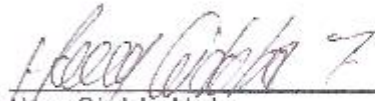
**INGENIERÍA DE SISTEMAS INFORMÁTICOS**

**POPAYÁN, NOVIEMBRE DE 2017**


**NOTA DE ACEPTACIÓN**

Popayán 2018

El director y los jurados del trabajo de grado en modalidad trabajo de Investigación denominado "DISEÑO E IMPLEMENTACIÓN DE UNA PLATAFORMA WEB PARA LA GESTIÓN DE LA INFORMACIÓN Y SEGUIMIENTO DE LOS TRABAJOS DE GRADO DE LA FACULTAD DE INGENIERÍA EN LA CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA", realizado por los estudiantes Andrés Felipe Vásquez y Octavio Andrés Cifuentes Dorado una vez revisado el informe final y aprobado la sustentación del mismo, autorizan para que se realicen los trámites concernientes para optar por el título profesional de Ingeniería de Sistemas informáticos.

  
Henry Córdoba Idrobo  
Jurado Evaluador

  
Gabriel Ángel Osorio Hoyos  
Jurado Evaluador

  
Julián Darío Bermúdez Trujillo  
Director

## **AGRADECIMIENTOS**

Primero agradecer a Dios por guiarme y permitirme culminar este trabajo con éxito. A mis padres por su apoyo durante este proceso, y a mi esposa e hijo por estar siempre ahí para mí y apoyarme hasta el final en este proyecto.

A mis compañeros y docentes universitarios por aportarme sus conocimientos.

Finalmente a nuestro director de trabajo de grado, por guiarnos en cada paso de este proceso, hasta culminar este proyecto.

### **Octavio Andrés Cifuentes Dorado**

En primer lugar, quiero agradecer a Dios por haber sido mi guía en el cumplimiento de esta meta. A mi madre por haber estado día a día ayudándome y dándome la fuerza necesaria cuando el camino se ponía difícil, por no desfallecer y por la fortaleza que me brindó en este camino

A mis compañeros y docentes universitarios, por todos los conocimientos aportados en mi construcción como ingeniero y persona.

Por último, agradezco a nuestro director de trabajo de grado, por su entrega incondicional, por su apoyo, por la guía, por sus invaluable conocimientos y por su ayuda para poder culminar con éxito esta etapa de mi vida.

### **Andrés Felipe Vásquez**

## CONTENIDO

RESUMEN .....	13
ABSTRACT .....	14
INTRODUCCIÓN .....	15
1. PLANTEAMIENTO DEL PROBLEMA .....	16
1.1 Pregunta de investigación.....	17
1.2 JUSTIFICACIÓN .....	17
1.2.1 Justificación técnica: .....	17
1.2.2 Justificación funcional: .....	18
1.2.3 Justificación económica-financiera.....	19
1.2.4 Justificación social: .....	19
1.3 OBJETIVOS .....	20
1.3.1 Objetivo General.....	20
1.3.2 Objetivos Específicos. ....	20
2. MARCO TEÓRICO. ....	21
2.1 MARCO CONTEXTUAL.....	21
2.2 ANTECEDENTES. ....	21
2.3 BASES TEÓRICAS.....	25
2.3.1 Metodologías Ágiles.....	25
2.3.3 Lenguajes de programación.....	28
2.4 METODOLOGÍA DE DESARROLLO DE SOFTWARE Y MARCO DE TRABAJO PARA LA GESTIÓN DE PROYECTO .....	31
2.4.1 Metodología Ágil XP (Xtreme Programming). ....	31
2.4.2 Scrum. ....	38
2.4.3 Patrón de Diseño (MVC).....	46
2.5 HERRAMIENTAS DE APOYO. ....	47
2.5.1 Xampp. ....	47
2.5.2 PhpMyAdmin. ....	47
2.5.3 Bitbucket.....	48
2.5.4 Javascript.....	49
2.5.5 Bootstrap. ....	49
2.5.6 JQuery.....	50
2.5.7 Php Mailer. ....	50
2.4.8 Apache (SERVIDOR). ....	50

2.4.9 Scrum Póker. ....	50
2.4.10 Atom. ....	51
2.6 GLOSARIO .....	52
<b>3. METODOLOGÍA.....</b>	<b>54</b>
<b>3.1 MARCO DE TRABAJO SCRUM.....</b>	<b>54</b>
3.1.1 Planificación del Sprint 0.....	54
3.1.2 Planificación del Sprint 1.....	56
3.1.3 Planificación del Sprint 2.....	56
3.1.4 Planificación del Sprint 3.....	57
3.1.5 Planificación Daily Scrum.....	57
3.1.6 Sprint Review.....	58
3.1.7 Retrospectiva.....	58
<b>3.2 XP (XTREME PROGRAMMING).....</b>	<b>59</b>
3.2.1 Planificación. ....	59
3.2.2 Diseño. ....	59
3.2.3 Codificación.....	59
3.2.4 Pruebas. ....	60
<b>3.3 ARQUITECTURA DEL SISTEMA.....</b>	<b>60</b>
<b>4. INGENIERÍA.....</b>	<b>62</b>
<b>4.1 IMPLEMENTACIÓN DEL SPRINT 0.....</b>	<b>62</b>
4.1.2 Diagrama de Base de Datos .....	62
4.1.3 Desarrollo de Agile Inception.....	64
<b>4.2 IMPLEMENTACIÓN DEL SPRINT 1.....</b>	<b>67</b>
4.2.1 Planificación. ....	67
4.2.2 Diseño. ....	101
4.2.3 Desarrollo.....	101
4.2.4 Pruebas. ....	103
4.2.5 Sprint Review.....	104
4.2.6 Retrospectiva.....	104
<b>4.3 IMPLEMENTACIÓN DEL SPRINT 2.....</b>	<b>105</b>
4.3.1 Planificación. ....	105
4.3.2 Diseño. ....	133
4.3.3 Desarrollo.....	133
4.3.4 Pruebas. ....	134

4.3.5 Sprint Review.....	136
4.3.6 Retrospectiva.....	136
<b>4.4 IMPLEMENTACIÓN DEL SPRINT 3.....</b>	<b>137</b>
4.4.1 Planificación. ....	138
4.4.2 Diseño. ....	169
4.4.3 Desarrollo.....	169
4.4.4 Pruebas. ....	170
4.4.5 Sprint Review.....	172
4.4.6 Retrospectiva.....	172
<b>5. RESULTADOS .....</b>	<b>173</b>
<b>6. CONCLUSIONES .....</b>	<b>176</b>
6.1 TRABAJO FUTURO. ....	177
<b>7. BIBLIOGRAFÍA.....</b>	<b>178</b>

## LISTA DE FIGURAS

Figura 1. Diagrama de Flujo .....	23
Figura 2. Diagrama de actividades por calles .....	24
Figura 3. Metodologías Ágiles. ....	25
Figura 4. Metodología Ágil XP. ....	32
Figura 5. Marco de trabajo Scrum. ....	38
Figura 6. Equipo de Scrum.....	39
Figura 7. Artefactos de Scrum.....	41
Figura 8. Eventos de Scrum.....	43
Figura 9. Xampp.....	47
Figura 10. PhpMyAdmin.....	48
Figura 11. Bitbucket.....	49
Figura 12. Scrum Póker.....	51
Figura 13. Atom.....	52
Figura 14. Retrospectiva Estrella de Mar.....	58
Figura 15. Arquitectura Cliente-Servidor.....	61
Figura 16. Diagrama de Base de Datos.....	64
Figura 17. Diseña tu caja.....	65
Figura 18. Interfaz Registrar Usuario .....	101
Figura 19. Prueba Funcional Interfaz Registrar Usuario .....	103
Figura 20. Prueba Funcional Registrar Usuario .....	104
Figura 21. Interfaz Asignar Fecha de Sustentación Final.....	133
Figura 22. Interfaz Asignar Fechas de sustentación Final .....	135
Figura 23. Interfaz Prueba de Asignar Fecha de Sustentación.....	135
Figura 24. Interfaz Ver detalles .....	136
Figura 25. Interfaz Actualizar Sustentación Final .....	136
Figura 26. Interfaz Crear Anteproyecto .....	169
Figura 27. Interfaz Crear anteproyecto .....	171
Figura 28. Interfaz Resultado Crear Anteproyecto.....	171



## LISTA DE TABLAS

Tabla 1. Scrum Master .....	54
Tabla 2. Development Team.....	55
Tabla 3. Development Team.....	55
Tabla 4. Product Owner .....	55
<i>Tabla 5. Retrospectiva del Sprint 0. ....</i>	<i>67</i>
<i>Tabla 6. Historia de Usuario Autenticar Usuario.....</i>	<i>68</i>
<i>Tabla 7. Task Card Crear Interfaces .....</i>	<i>69</i>
Tabla 8. Task Card: Conexión al servidor.....	70
Tabla 9. Task Card: Validación de la información.....	70
Tabla 10. Task Card: Testing .....	71
Tabla 11. Task Card: Integración .....	72
Tabla 12. Historia de usuario Registrar usuario.....	72
<i>Tabla 13. Task Card: Crear Interfaces.....</i>	<i>74</i>
Tabla 14. Task Card: Conexión al servidor .....	74
Tabla 15. Task Card: Validación de datos ingresados por el usuario .....	75
<i>Tabla 16. Task Card: Crear nuevo registro .....</i>	<i>76</i>
<i>Tabla 17. Task card: Testing.....</i>	<i>76</i>
<i>Tabla 18. Task Card: Integración.....</i>	<i>77</i>
<i>Tabla 19. Historia de usuario Seleccionar Archivo.....</i>	<i>78</i>
<i>Tabla 20. Task Card: Crear Interfaces.....</i>	<i>79</i>
<i>Tabla 21. Task Card: Conexión al servidor.....</i>	<i>80</i>
<i>Tabla 22. Task card: Verificación Del documento.....</i>	<i>80</i>
Tabla 23. Task Card: Crear seleccionar archivo en la plataforma .....	81
<i>Tabla 24. Task Card: Testing .....</i>	<i>82</i>
<i>Tabla 25. Task Card: Integración.....</i>	<i>82</i>
<i>Tabla 26. Historia de usuario Consultar información Estudiante.....</i>	<i>83</i>
<i>Tabla 27. Task Card: Crear Interfaces.....</i>	<i>84</i>
<i>Tabla 28. Task Card: Conexión al servidor.....</i>	<i>85</i>
<i>Tabla 29. Task Card: Consultar información.....</i>	<i>86</i>
<i>Tabla 30. Task Card: Testing .....</i>	<i>86</i>
<i>Tabla 31. Task Card: Integración.....</i>	<i>87</i>

<b>Tabla 32. Historia de usuario Cambiar Contraseña .....</b>	<b>88</b>
<b>Tabla 33. Task Card: Crear Interfaces .....</b>	<b>89</b>
<b>Tabla 34. Task Card: Conexión al servidor.....</b>	<b>89</b>
<b>Tabla 35. Task Card: Actualización de registro solicitado por el usuario .</b>	<b>90</b>
<b>Tabla 36. Task Card: Testing .....</b>	<b>91</b>
<b>Tabla 37. Task Card: Integración.....</b>	<b>91</b>
<b>Tabla 38. Historia de usuario: Consultar Reportes.....</b>	<b>92</b>
<b>Tabla 39. Task Card: Crear Interfaces.....</b>	<b>93</b>
<b>Tabla 40. Task Card: Conexión al servidor.....</b>	<b>94</b>
<b>Tabla 41. Task card: Generar y visualizar reportes .....</b>	<b>94</b>
<b>Tabla 42. Task Card: Testing .....</b>	<b>95</b>
<b>Tabla 43. Task Card: Integración.....</b>	<b>96</b>
<b>Tabla 44. Historia de Usuario Consultar Información Miembro Comité.....</b>	<b>96</b>
<b>Tabla 45. Task Card: Crear Interfaces .....</b>	<b>97</b>
<b>Tabla 46. Task Card: Conexión al servidor.....</b>	<b>98</b>
<b>Tabla 47. Task Card: Consultar información.....</b>	<b>99</b>
<b>Tabla 48. Task card: Testing.....</b>	<b>99</b>
<b>Tabla 49. Task Card: Integración.....</b>	<b>100</b>
<b>Tabla 50. Evento Programado Registro de Usuario.....</b>	<b>101</b>
<b>Tabla 51. Prueba Funcional Registro de Usuario.....</b>	<b>103</b>
<b>Tabla 52. Retrospectiva del Sprint 1. ....</b>	<b>105</b>
<b>Tabla 53. Historia de usuario Consultar información Jurado .....</b>	<b>106</b>
<b>Tabla 54. Task Card: Crear Interfaces.....</b>	<b>107</b>
<b>Tabla 55. Task card: Conexión al servidor .....</b>	<b>107</b>
<b>Tabla 56. Task Card: Consultar información.....</b>	<b>108</b>
<b>Tabla 57. Task Card: Testing .....</b>	<b>109</b>
<b>Tabla 58. Task Card: Integración.....</b>	<b>109</b>
<b>Tabla 59. Historia de usuario Recibir alerta.....</b>	<b>110</b>
<b>Tabla 60. Task Card: Crear Interfaces.....</b>	<b>111</b>
<b>Tabla 61. Task Card: Conexión al servidor.....</b>	<b>112</b>
<b>Tabla 62. Task Card: Crear Sistema de Alertas.....</b>	<b>112</b>
<b>Tabla 63. Task Card: Testing .....</b>	<b>113</b>

<b>Tabla 64. Task Card: Integración.....</b>	<b>113</b>
<b>Tabla 65. Historia de usuario Consultar información Docente.....</b>	<b>114</b>
<b>Tabla 66. Task Card: Crear Interfaces.....</b>	<b>116</b>
<b>Tabla 67. Task Card: Conexión al servidor.....</b>	<b>116</b>
<b>Tabla 68. Task Card: Consultar información.....</b>	<b>117</b>
<b>Tabla 69. Task Card: Testing.....</b>	<b>117</b>
<b>Tabla 70. Task Card: Integración.....</b>	<b>118</b>
<b>Tabla 71. Historia de usuario Rechazar proyecto.....</b>	<b>119</b>
<b>Tabla 72. Task Card: Crear Interfaces.....</b>	<b>120</b>
<b>Tabla 73. Task Card: Conexión al Servidor.....</b>	<b>120</b>
<b>Tabla 74. Task Card: Cambiar Estado de un proyecto.....</b>	<b>121</b>
<b>Tabla 75. Task Card: Testing.....</b>	<b>122</b>
<b>Tabla 76. Task Card: Integración.....</b>	<b>122</b>
<b>Tabla 77. Historia de usuario Asignar Jurados.....</b>	<b>123</b>
<b>Tabla 78. Task Card: Crear Interfaces.....</b>	<b>124</b>
<b>Tabla 79. Task Card: Conexión al servidor.....</b>	<b>125</b>
<b>Tabla 80. Task Card: Crear asignar jurado.....</b>	<b>125</b>
<b>Tabla 81. Task Card: Testing.....</b>	<b>126</b>
<b>Tabla 82. Task Card: Integración.....</b>	<b>127</b>
<b>Tabla 83. Historia de usuario Asignar Fecha de Sustentación.....</b>	<b>127</b>
<b>Tabla 84. Task Card: Crear Interfaces.....</b>	<b>129</b>
<b>Tabla 85. Task Card: Conexión al Servidor.....</b>	<b>129</b>
<b>Tabla 86. Task Card: Consultar Cronograma.....</b>	<b>130</b>
<b>Tabla 87. Task Card: Crear fecha de sustentación.....</b>	<b>131</b>
<b>Tabla 88. Task Card: Testing.....</b>	<b>131</b>
<b>Tabla 89. Task Card: Integración.....</b>	<b>132</b>
<b>Tabla 90. Evento Asignar Fecha de Sustentación.....</b>	<b>133</b>
<b>Tabla 91. Prueba Funcional Asignar Sustentación Final.....</b>	<b>134</b>
<b>Tabla 92. Retrospectiva del Sprint 2.....</b>	<b>137</b>
<b>Tabla 93. Historia de usuario Crear Jurado.....</b>	<b>138</b>
<b>Tabla 94. Task Card Crear Interfaces.....</b>	<b>139</b>
<b>Tabla 95. Task Card: Conexión al servidor.....</b>	<b>140</b>

<b>Tabla 96. Task Card: Crear jurado.....</b>	<b>140</b>
<b>Tabla 97. Task Card: Testing .....</b>	<b>141</b>
<b>Tabla 98. Task Card: Integración.....</b>	<b>141</b>
<b>Tabla 99. Historia de usuario Recuperar Contraseña.....</b>	<b>142</b>
<b>Tabla 100. Task Card: Crear Interfaces.....</b>	<b>143</b>
<b>Tabla 101. Task Card: Conexión al servidor.....</b>	<b>144</b>
<b>Tabla 102. Task Card: Crear recuperación de contraseña .....</b>	<b>144</b>
<b>Tabla 103. Task Card: Testing .....</b>	<b>145</b>
<b>Tabla 104. Task Card: Integración.....</b>	<b>146</b>
<b>Tabla 105. Historia de usuario Crear Anteproyecto.....</b>	<b>146</b>
<b>Tabla 106. Task Card: Crear Interfaces.....</b>	<b>147</b>
<b>Tabla 107. Task Card: Conexión al Servidor .....</b>	<b>148</b>
<b>Tabla 108. Task Card: Crear Anteproyecto.....</b>	<b>149</b>
<b>Tabla 109. Task Card: Testing .....</b>	<b>149</b>
<b>Tabla 110. Task Card: Integración.....</b>	<b>150</b>
<b>Tabla 111. Historia de usuario Crear Proyecto.....</b>	<b>151</b>
<b>Tabla 112. Task Card: Crear Interfaces.....</b>	<b>152</b>
<b>Tabla 113. Task Card: Conexión al Servidor .....</b>	<b>153</b>
<b>Tabla 114. Task Card: Crear Proyecto .....</b>	<b>153</b>
<b>Tabla 115. Task Card: Testing .....</b>	<b>154</b>
<b>Tabla 116. Task Card: Integración.....</b>	<b>155</b>
<b>Tabla 117. Historia de usuario Asignar Concepto .....</b>	<b>155</b>
<b>Tabla 118. Task Card: Crear Interfaces.....</b>	<b>157</b>
<b>Tabla 119. Task Card: conexión al servidor .....</b>	<b>157</b>
<b>Tabla 120. Task Card: Crear concepto.....</b>	<b>158</b>
<b>Tabla 121. Task Card: Testing .....</b>	<b>159</b>
<b>Tabla 122. Task Card: Integración.....</b>	<b>159</b>
<b>Tabla 123. Historia de usuario Crear Director Proyecto.....</b>	<b>160</b>
<b>Tabla 124. Task Card: Crear Interfaces.....</b>	<b>161</b>
<b>Tabla 125. Task Card: Conexión al Servidor .....</b>	<b>162</b>
<b>Tabla 126. Task Card: Crear Director de Proyecto.....</b>	<b>162</b>
<b>Tabla 127. Task Card: Testing .....</b>	<b>163</b>

<b>Tabla 128. Task card: Integración .....</b>	<b>164</b>
<b>Tabla 129. Historia de usuario Crear Revisión.....</b>	<b>164</b>
<b>Tabla 130. Task Card: Crear Interfaces.....</b>	<b>166</b>
<b>Tabla 131. Task Card: Conexión al servidor.....</b>	<b>166</b>
<b>Tabla 132. Task Card: Crear concepto.....</b>	<b>167</b>
<b>Tabla 133. Task Card: Testing .....</b>	<b>168</b>
<b>Tabla 134. Task Card: Integración.....</b>	<b>168</b>
<b>Tabla 135. Evento Programado Crear Anteproyecto. ....</b>	<b>169</b>
<b>Tabla 136. Pruebas funcionales Crear Anteproyecto. ....</b>	<b>170</b>
<b>Tabla 137. Retrospectiva del Sprint 3. ....</b>	<b>172</b>

## RESUMEN

Éste proyecto está orientado a la gestión de proyectos de grado y pasantías de la facultad de Ingenierías de la Corporación Universitaria Autónoma del Cauca. Su desarrollo fue guiado bajo el marco de trabajo Scrum y algunas prácticas de metodologías ágiles (Programación extrema) que comprendieron las 4 fases de desarrollo: planeación, diseño, desarrollo y pruebas. El objetivo principal del producto del proyecto es el desarrollo de una plataforma web que facilite la gestión de la información de los proyectos y agilice el seguimiento de estos mismos. Por otro lado, se programan alarmas para recordar a los estudiantes y docentes del comité las entregas oportunas de los avances de acuerdo al cronograma establecido, y basándose en la norma 0047 que rige los trabajos de grado, logrando así eficiencia y eficacia en tiempos, y haciendo más simple el trabajo de la institución.

Palabras Claves: Scrum, metodologías ágiles, plataforma web, trabajos de grado.

## **ABSTRACT**

This project is oriented to the management of grade projects and internships of the faculty of engineering of the Corporación Universitaria Autónoma del Cauca. Its development was guided under the framework of Scrum and some practices of agile methodologies (Extreme Programming) which included the 4 phases of development: planning, design, development and tests. The main objective of the project's product is the development of a web platform that facilitates the management of project information and speeds up the tracking of these. On the other hand, alarms are programmed to remind the students and teachers of the committee of timely deliveries of the progress according to the established schedule, and based on the rule that governs grade works, thus achieving efficiency and effectiveness in time, and making the work of the institution simpler.

Keywords: Scrum, agile methodologies, web platform, grade Works.

## INTRODUCCIÓN

El objeto de estudio de este proyecto es la Facultad de Ingenierías de la Corporación Universitaria Autónoma del Cauca, donde se llevan a cabo todos los procesos concernientes a las carreras de ingeniería electrónica y sistemas, algunos de estos aún se hacen de manera manual y otros por medios informáticos que les facilita el manejo a los funcionarios encargados de dichos procesos.

La facultad de ingeniería ha logrado establecer con el tiempo y por medio de ayudas informáticas y manuales, el proceso de proyectos de grado, toda la gestión de su información y el seguimiento de los mismos, pero esta manera de hacerlo ha traído consigo debilidades en cuanto a tiempos de respuesta para los estudiantes y miembros de comité, retrasos a la hora de hacer determinadas consultas y problemas entre los diferentes actores implicados en un proceso tan primordial en una carrera universitaria, logrando así el incumplimiento de la norma que rige los proyectos de grado en la Universidad. Para aportar una solución tecnológica al problema anteriormente mencionado se propuso el desarrollo de una plataforma web con sus etapas de planificación, diseño, desarrollo y pruebas para la facultad de ingeniería de la Corporación Universitaria Autónoma del Cauca.

Y aunque este proceso ya funciona actualmente, éste aporte tecnológico le brinda la posibilidad de mejorar y hacer de manera eficiente y controlada todo lo que trae consigo, dando solución al problema antes mencionado y sentando un precedente para posibles y futuras mejoras en otras facultades.



## 1. PLANTEAMIENTO DEL PROBLEMA

Las universidades deben manejar y gestionar la información referente a los trabajos de grado de sus estudiantes de una manera clara, entendible y rápida que permita a las partes involucradas un fácil acceso, buena comunicación y un entorno de trabajo más productivo, que se traduce en beneficios para estudiantes, docentes, miembros del comité de investigación, directores de trabajos de grado, secretarías de la facultad, oficina de registro y control y en general a todos los participantes del proceso.

Para citar un ejemplo, la Universidad como EAFIT (sistema sirena), ha utilizado herramientas informáticas para este tipo de procesos, logrando una dinámica mucho más efectiva, con mejores resultados y mejores ayudas al momento de realizar las actividades y cumplir con los ítems que exigen los proyectos grado.

Teniendo en cuenta lo anterior, se ha identificado que en la facultad de ingeniería de la Corporación universitaria Autónoma del Cauca, los procesos de gestión y seguimiento de trabajos de grado se apoyan con software de ofimática (Microsoft Excel) que sirve para el registro de información, pero no para obtener indicadores adecuados que permitan tomar decisiones futuras; solamente se limita a lo que esta herramienta ofrezca.

Por otra parte, la asignación en cantidad de trabajos de grado, de directores y revisores por encima de lo establecido por la resolución 0047 conlleva a que resulte difícil el cumplimiento en las fechas por parte de los mismos, incluso si estos roles son desempeñados por personas externas a la institución.

Luego de las encuestas realizadas a docentes miembros del comité y estudiantes en proceso de trabajo de grado se encontró que es común ver que en un 95% las revisiones y posteriores correcciones a realizar que deben ser enviadas por el comité se demoran más de los 15 días hábiles que se encuentran estipulados en la resolución 0047, y además se evidencia en más de un 60% el desinterés del estudiante al momento de realizar y entregar las respectivas correcciones, causando con esto, retrasos en la dinámica del desarrollo de un trabajo de grado. Otras de las grandes dificultades que se

presentan son: la excesiva asignación de trabajos de grado a un mismo docente, sabiendo que el límite es tres (3) con el rol de jurado y dos (2) con el rol de director; y el hecho de contar con una persona (en este caso el coordinador de los programas de la facultad) la cual ha creado un sistema en Excel con convenciones definidos por esta misma y que solo son de su conocimiento, por medio del cual esta lleva el control del proceso; lo cual trae consigo una interrupción del mismo en el momento en el que ésta persona se encuentre ausente por distintos motivos.

Si éste proceso se sigue ejecutando de la misma manera, es probable que continúe funcionando de forma inadecuada como ya se mencionó anteriormente, esto no quiere decir que no funcione, es solo que no tendrá la misma eficiencia y eficacia que se obtendrá si se utiliza la plataforma web. Al insistir en el mecanismo actual, la facultad recaerá en el incumplimiento de tiempos de entrega y seguirá pasando por alto la norma 0047 que establece lineamientos para el proceso de trabajos de grado, por lo que se hace necesaria la implementación de metodologías ágiles que permitan optimizar tiempo y dinero, que sería a futuro el problema más grave que podría traer el seguir usando el método actual.

### **1.1 Pregunta de investigación.**

¿Cómo se puede mejorar el proceso de gestión y seguimiento de trabajos de grado en la facultad de ingenierías, a través del desarrollo de una solución informática?

## **1.2 JUSTIFICACIÓN**

### **1.2.1 Justificación técnica:**

Con el desarrollo de este proyecto se pretende brindarle a la facultad de ingenierías de la corporación universitaria autónoma del cauca una solución tecnológica que vaya acorde a las necesidades que presenta actualmente en el tema de trabajos de grado, una plataforma informática que haga una sistematización de los procesos internos y un seguimiento oportuno y bien

programado de los trabajos de grado presentados por los estudiantes, para de esta manera poder optimizar tiempo de respuesta y mejorar la calidad de educación prestada dándole solución a la dificultad actual de la facultad en cuanto a trabajos de grado se refiere.

De esta manera, se ponen en práctica los conocimientos desarrollados en ingeniería del software, programación, materias afines a las matemáticas y bases de datos, proponiendo así una solución óptima y eficaz para mejorar el proceso de gestión de información y seguimiento de proyectos de grado de la facultad de Ingeniería de la Corporación Universitaria Autónoma del Cauca.

### **1.2.2 Justificación funcional:**

Con el desarrollo de la plataforma web se busca hacer la gestión de la información pertinente a trabajos de grado, es decir, se propone gestionar paso a paso y de manera ordenada el proceso que se lleva a cabo internamente en cada trabajo de grado, cronogramas de entrega, con fechas estipuladas con anterioridad, y cronogramas de reuniones de comité ya establecidas para todo un semestre, fechas de radicación y alertas continuas para recordar estas fechas. Además de hacer alertas por perfiles, es decir, alertar tanto a comité como a estudiantes que tienen un compromiso con fecha establecida para entrega ya sea de correcciones o de proyectos para sustentación. Además, es una plataforma que permite crear un perfil propio que se podrá revisar desde cualquier dispositivo que se tenga a la mano y estará siendo avisado no solo por alertas programadas sino por medio de correos para que la entrega del trabajo se haga de manera oportuna. Además se tendrá la oportunidad de escoger proyectos que sean propuestos por el profesorado para llevarlos a cabo como trabajos de grado. Cada estudiante podrá mirar en qué estado se encuentra su proyecto, al igual que los miembros del comité podrán ver cuantos proyectos están en marcha y en qué estado se encuentran de manera que el seguimiento es mucho más eficaz.

De esta forma, al implementarse la plataforma web, traerá beneficios a todos los actores implicados, tales como la eliminación de procesos manuales como

(manejo de la plataforma Excel, convenciones creadas por el secretario, redacción de correos para enviar proyectos a comité, creación de reportes manuales, búsqueda manual de documentos, creación de cronogramas, asignación de jurados) y el control exacto de otros como el seguimiento puntual de cada proyecto y la oportuna entrega de los mismos que en últimas se traduce en ahorro de tiempo y dinero para la facultad como para el estudiante. Además, se centraliza la información, se hace más fácil las consultas y hay más seguridad en el manejo de datos.

Los tiempos de respuesta serán reducidos, la plataforma es más organizada y rápida que el proceso actual, no habrá retrasos en días puesto que esta se ajusta a los lineamientos propuestos por la norma 0047. Solucionará el problema para los usuarios implicados, los estudiantes recibirán las correcciones sugeridas a sus trabajos de grado dentro del tiempo establecido por la norma y así mismo tendrán que enviar estas en los días estipulados favoreciendo de esta manera al docente del comité, ya que habrá un flujo continuo en el proceso y no se verá interrumpido por ninguna de las partes por no cumplir con los tiempos. De esta manera, los tiempos de respuesta se reducen con el uso de la plataforma web de manera notable.

### **1.2.3 Justificación económica-financiera**

Esta plataforma web traerá beneficios a las partes implicadas tales como: ahorro en costos de transporte para los estudiantes y docentes puesto que ya podrán informarse desde cualquier parte donde se encuentren, además optimización de tiempo ya que se reduce horas de trabajo al personal involucrado en el proceso, y estas horas podrán ser destinadas a otras tareas. Por otro lado, encontramos la reducción de costos en insumos de papelería que podrán ser ahorrados o destinados a otros gastos.

### **1.2.4 Justificación social:**

Los trabajos de grado de la universidad autónoma del cauca, están regidos bajo la Resolución No 0047 DE 2012, la plataforma y su implementación ayudará a que esta norma sea cumplida a cabalidad y que los estudiantes a

puerta de graduarse como el comité, tengan una dinámica más activa y responsable en esta etapa final de la carrera.

### **1.3 OBJETIVOS**

#### **1.3.1 Objetivo General.**

Desarrollar una plataforma web para la gestión del proceso vigente de trabajos de grado de la facultad de Ingeniería de la Corporación Universitaria Autónoma del Cauca.

#### **1.3.2 Objetivos Específicos.**

- Disminuir los procesos manuales de la gestión de trabajos de grado de la facultad de ingeniería.
- Simplificar el flujo de información dentro del proceso de trabajos de grado de la facultad de ingeniería.
- Reducir los tiempos de respuesta que se maneja en el proceso actual de trabajos de grado.

## **2. MARCO TEÓRICO.**

### **2.1 MARCO CONTEXTUAL.**

El proyecto será desarrollado en la corporación universitaria autónoma el cauca, ubicada en la ciudad de Popayán, departamento del Cauca. Está dirigido para la facultad de ingeniería de esta universidad, específicamente para los actores implicados en el proceso de trabajos de grado tales como estudiantes en trabajo de grado y en busca de este mismo, docentes miembros del comité y funcionarios encargados de la gestión y manejo de este proceso. El comité es el encargado de revisar y aprobar dichos trabajos presentados por los estudiantes, y los funcionarios son aquellos que intervienen en el proceso llevado a cabo entre el estudiante y el comité, por tal razón este proyecto se encamina hacia estos actores y en este lugar con el fin de mejorar el proceso que se maneja actualmente.

### **2.2 ANTECEDENTES.**

En la actualidad la gestión de la información en la Corporación Universitaria Autónoma del Cauca, se hace por medio de archivos planos, dificultando el manejo de esta misma. Para satisfacer las necesidades de los usuarios con la elaboración de una plataforma web, se debe hacer un estudio de elementos fundamentales como: marco de trabajo, metodología software y lenguajes de programación.

Una institución que busca facilitar dicho proceso es la Fundación Universitaria de Popayán que se encuentra en una etapa prematura del desarrollo de una herramienta informática o hasta de un desarrollo software. Otras instituciones como la Universidad del Cauca, UniComfacauca, Fundación Universitaria María Cano, Universidad Cooperativa de Colombia, Colegio Mayor del Cauca, ubicadas en la ciudad de Popayán no cuentan con una integración de sus actividades de proyecto de grado a nivel software y todos estos procesos son llevados de manera manual, haciendo que estos no se hagan de la manera más eficiente.

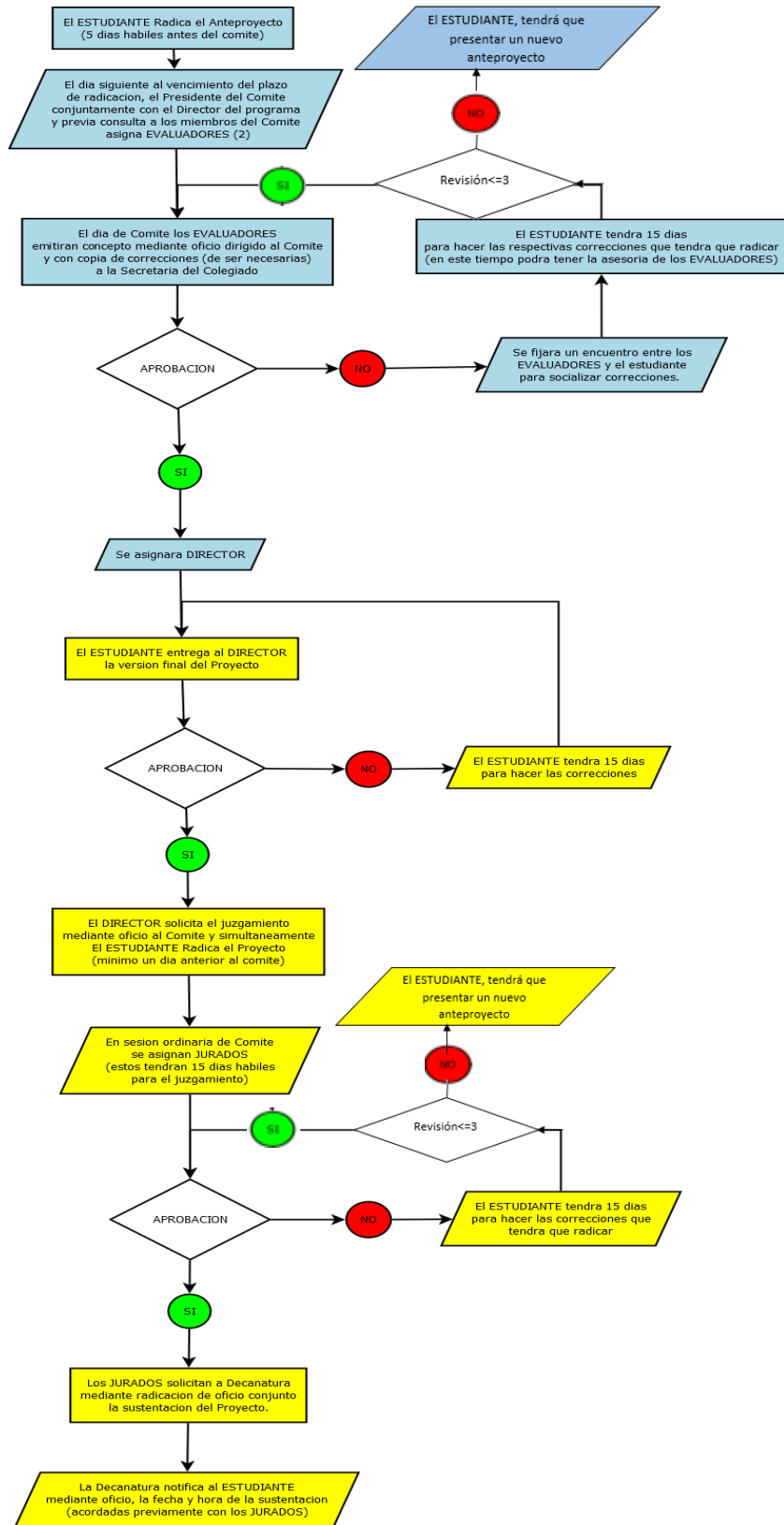
En el año 2015, se dió a conocer en la Corporación Universitaria Autónoma del Cauca un proyecto relacionado y que abrió puertas a la investigación en este campo, pero que no se llevó a cabo su implementación porque no se hizo la correcta elicitación de requisitos con cada uno de los involucrados, por este motivo, el desarrollo software no se adaptó a las necesidades de la institución.

Se considera otros trabajos relacionados con el de éste proyecto; OrangeScrum para la gestión de proyectos (software libre/ Open Source), que es una aplicación web básica para la gestión y colaboración de proyectos en pequeñas y medianas empresas para Colombia Ecuador, Chile, Perú y otros países de América Latina, este aplicativo permite crear y administrar proyectos, equipos de trabajo, documentos y tareas, además de una comunicación entre los miembros del proyecto por medio web y móvil. [1]

Otro trabajo relacionado es RedMine Flexible Project Management, que “es una plataforma web que gestiona proyectos e incluye un sistema de incidencias, seguimiento de errores, calendario de actividades, diagrama de Gantt para la representación de los proyectos a manera visual, repositorio de versiones e integración de correos electrónicos”. [2]

### **Flujo Actual de Trabajo.**

En la siguiente figura se muestra el proceso que actualmente sigue la facultad de ingeniería de la institución:



**Figura 1. Diagrama de Flujo [Fuente. Corporación Universitaria Autónoma del Cauca.]**



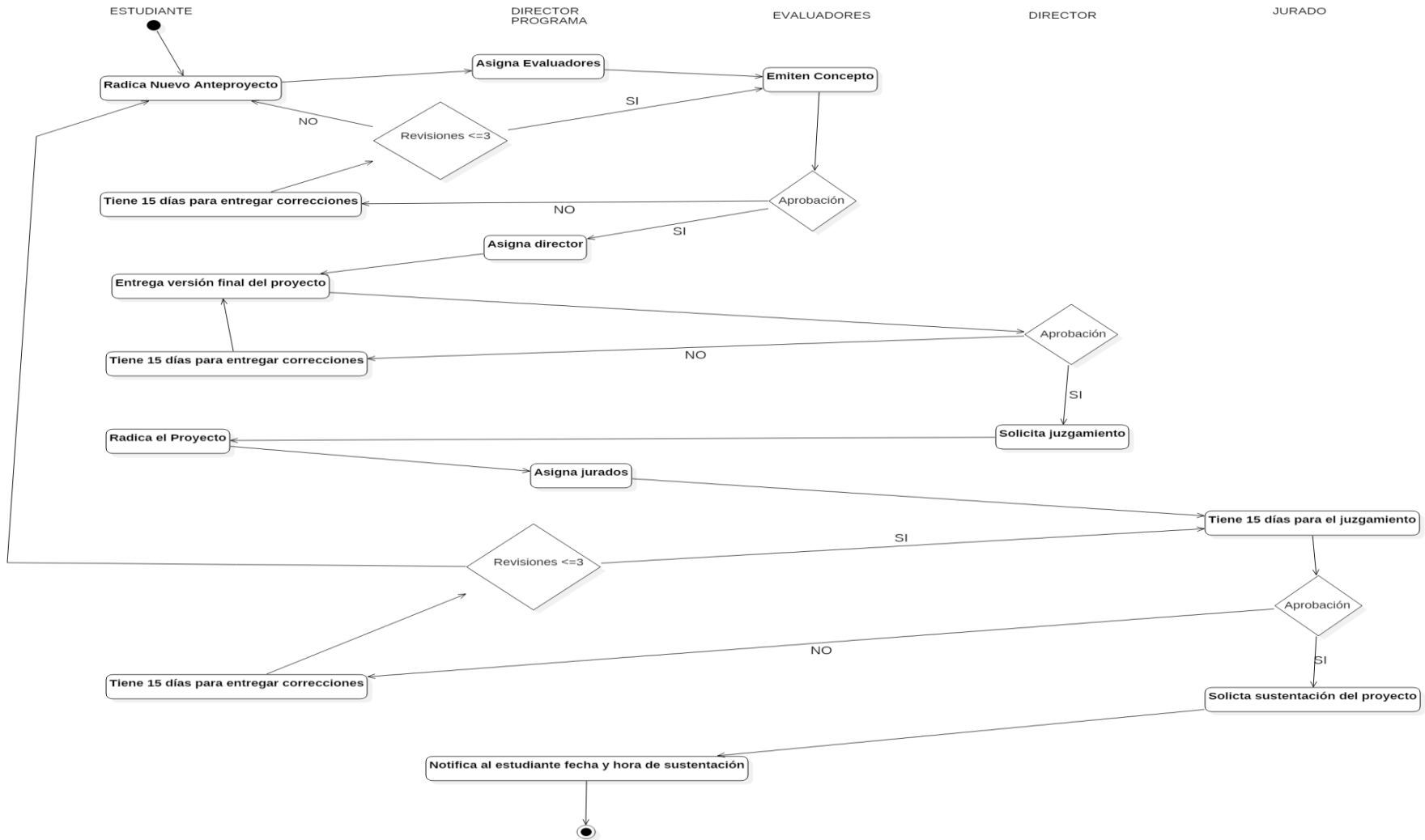


Figura 2. Diagrama de actividades por calles [Fuente Propia]

## 2.3 BASES TEÓRICAS.

### 2.3.1 Metodologías Ágiles.

“En febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace el término “ágil” aplicado al desarrollo de software. En esta reunión participan un grupo de 17 expertos de la industria del software, incluyendo algunos de los creadores o impulsores de metodologías de software. Su objetivo fue esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas.

Tras esta reunión se creó “*The Agile Alliance*”, una organización, sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida es fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”.” [3]



**Figura 3. Metodologías Ágiles.**

[4]

### **2.3.1.1 EL Manifiesto Ágil.**

“En 2001 se crea el Manifiesto por el desarrollo ágil de software, documento en el que se acuerdan cuatro principios básicos para el desarrollo de software, que establece prioridades y marca diferencias de fondo frente a los sistemas tradicionales: individuos e interacciones, por encima de procesos y herramientas; software funcionando, por encima de documentación extensiva; colaboración con el cliente, por encima de negociación contractual; y respuesta ante el cambio, por encima de seguir un plan.

Los principios que dan origen al manifiesto implican la satisfacción del cliente mediante entregas tempranas y continuas de software que funcione; requerimientos cambiantes en cualquier etapa del proyecto; participación activa del cliente; simplicidad; equipos de desarrollo motivados y auto-organizados; comunicación efectiva; auto inspecciones; y adaptación.” [5]

### **2.3.1.2 Valores.**

- “Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas. La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades.
- Desarrollar software que funciona más que conseguir una buena documentación. La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar un decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- La colaboración con el cliente más que la negociación de un contrato. Se propone que exista una interacción constante entre el cliente y el equipo

de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.

- Responder a los cambios más que seguir estrictamente un plan. La habilidad de responder a los cambios que puedan surgir a los largo del proyecto determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.” [6]

### **2.3.1.3 Principios.**

- “La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- El software que funciona es la medida principal de progreso.
- Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.

- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.” [3]

### **2.3.2 Kanban.**

“El Kanban es un sistema de gestión del trabajo en curso, que sirve principalmente para asegurar una producción continua y sin sobrecargas en el equipo de producción multimedia. El Kanban es un sistema de gestión donde se produce exactamente aquella cantidad de trabajo que el sistema es capaz de asumir. El Kanban es un sistema de trabajo *“just in time”*, lo que significa que evita sobrantes innecesarios de stock, que en la gestión de proyectos multimedia equivale a la inversión innecesaria de tiempo y esfuerzo en lo que no necesitaremos y evita sobrecargar al equipo”. [7]

### **2.3.3 Lenguajes de programación**

**2.3.3.1 Php:** HyperText Preprocessor, Es un lenguaje interpretado de alto nivel, embebido en html, para la realización de páginas web. A diferencia de otros lenguajes de programación, éste se ejecuta por el lado del servidor.

Algunas características son:

“

- Es un lenguaje libre.
- Tiene una extensa documentación oficial en varios idiomas
- Existen multitud de extensiones: para conectar con bases de datos, para manejo de sockets, para generar documentos PDF, para generar dinámicamente páginas en Flash.
- Al ejecutarse en el servidor, los programas PHP lo pueden usar todo tipo de máquinas con todo tipo de sistemas operativos.

- En caso de que un cliente falle (por error hardware, virus, etc) se puede seguir usando el sistema desde otro cualquiera que tenga un navegador web con conexión al servidor.” [8]

**2.3.3.2 Html5:** HyperText Markup Language, “HTML5 es un lenguaje markup, usado para estructurar y presentar el contenido para la web. Es uno de los aspectos fundamentales para el funcionamiento de los sitios, pero no es el primero. Es de hecho la quinta revisión del estándar que fue creado en 1990. A fines del año pasado, la W3C la recomendó para transformarse en el estándar a ser usado en el desarrollo de proyectos venideros. Con HTML5, tenemos otras posibilidades para explotar usando menos recursos.

Una de las más importantes novedades está relacionada con la inserción de multimedia en los sitios web, que ahora contarán con etiquetas HTML especiales para poder ser incluidos.” [9]

**2.3.3.3 MySQL:** “Es un sistema de gestión de bases de datos relacional. MySQL es un software de código abierto.

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- El servidor soporta mensajes de error en distintas lenguas.” [10]

#### **2.3.3.4 Framework**

“El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de

visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírseles. En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.” [11]

#### **2.3.3.5 Yii**

“Yii es un framework PHP basado en componentes de alta performance para desarrollar aplicaciones Web de gran escala. El mismo permite la máxima reutilización en la programación web y puede acelerar el proceso de desarrollo”. [12]

**2.3.3.6 CSS:** Cascading Style Sheets, “es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo.” [13]

#### **2.3.3.7 BPM**

“Es un conjunto de métodos, herramientas y tecnologías utilizados para diseñar, representar, analizar y controlar procesos de negocio

operacionales. BPM es un enfoque centrado en los procesos para mejorar el rendimiento que combina las tecnologías de la información con metodologías de proceso y gobierno. BPM es una colaboración entre personas de negocio y tecnólogos para fomentar procesos de negocio efectivos, ágiles y transparentes.

Con BPM: \_ Los directores de negocio pueden, de forma más directa, medir, controlar y responder a todos los aspectos y elementos de sus procesos operacionales. \_ Los directores de tecnologías de la información pueden aplicar sus habilidades y recursos de forma más directa en las operaciones de negocio. \_ La dirección y los empleados de la organización pueden alinear mejor sus esfuerzos y mejorar la productividad y el rendimiento personal. \_ La empresa, como un todo, puede responder de forma más rápida a cambios y desafíos a la hora de cumplir sus fines y objetivos.” [14]

#### **2.3.4 Plataforma Web.**

“Cuando nos referimos a una plataforma web, estamos diciendo que nuestro sistema funcionará desde un navegador web, independientemente del sistema operativo (SO) que se esté utilizando, esto tiene la ventaja de que no es necesario crear el mismo software para cada tipo de sistema operativo.

Plataforma web podría hacer referencia al conjunto de programas y sistemas que hacen que un sitio web esté en funcionamiento: el servidor web, la base de datos, el intérprete de programación”. [15]

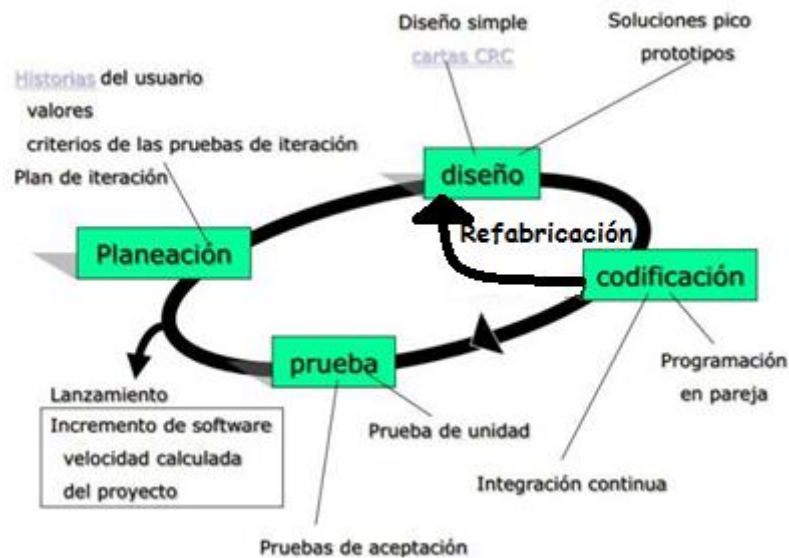
## **2.4 METODOLOGÍA DE DESARROLLO DE SOFTWARE Y MARCO DE TRABAJO PARA LA GESTIÓN DE PROYECTO**

### **2.4.1 Metodología Ágil XP (Xtreme Programming).**

Es una metodología ágil, que se basa en las buenas relaciones entre el grupo de trabajo, esto lo hace con el fin de tener éxito en el desarrollo del software, brindando un ambiente apropiado para trabajar. Un punto importante de XP es



la retroalimentación que existe entre el equipo de desarrollo y el cliente con diálogos para obtener soluciones simples. [16]



**Figura 4. Metodología Ágil XP.**

[17]

#### 2.4.1.1 Valores.

- **Comunicación:** “La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.” [18]
- **Simplicidad:** “XP, como metodología ágil, apuesta a la sencillez, en su máxima expresión. Sencillez en el diseño, en el código, en los procesos, etc. La sencillez es esencial para que todos puedan entender el código, y se trata de mejorar mediante recodificaciones continuas.” [18]
- **Retroalimentación:** “El cliente debe brindar retroalimentación de las funciones desarrolladas, de manera de poder tomar sus comentarios para la próxima iteración, y para comprender, cada vez más, sus

necesidades. Los resultados de las pruebas unitarias son también una retroalimentación permanente que tienen los desarrolladores acerca de la calidad de su trabajo.” [18]

- **Coraje:** “Cuando se encuentran problemas serios en el diseño, o en cualquier otro aspecto, se debe tener el coraje suficiente como para encarar su solución, sin importar que tan difícil sea. Si es necesario cambiar completamente parte del código, hay que hacerlo, sin importar cuanto tiempo se ha invertido previamente en el mismo.” [18]

#### 2.4.1.2 Fases.

- **Fase I. Planificación del proyecto:** Se realizan reuniones entre cliente y el equipo de desarrollo, para determinar el objetivo del proyecto.
  - ✓ **Las Historias de Usuario:** Es una técnica que se usa para desglosar y explicar más a detalle los requisitos del software, es decir, son unas tarjetas en las cuales se describen las características que el cliente desea que tenga el desarrollo. Cada una de las historias de usuario es lo necesariamente comprensible para así poder ser implementados en poco tiempo. Si se trata de tiempo, las historias de usuario deben ser realizadas y entregadas entre una (1) y tres (3) semanas. [6]
  - ✓ **Release planning (Plan de Entregas):** Luego de haber definido las historias de usuario, se debe crear un plan de publicaciones, en el cual se indican las historias de usuario creadas por versión y fecha de publicación de versiones. Además este release es la planificación en la cual el cliente y los desarrolladores establecen los tiempos y la prioridad de las historias que se van a implementar por cada versión.  
Cuando ya se tenga el release plan, hay que tener bien claros estos factores: Los objetivos a cumplir, el tiempo que se tardará el desarrollo de cada versión, la cantidad de integrantes del desarrollo y por último como se va a evaluar el trabajo. [19]

- ✓ **Iteraciones:** Son de una duración máxima de tres (3) semanas. Al comienzo de cada iteración los clientes deben seleccionar las historias de usuario definidas en el "Release planning" que serán implementadas. También se seleccionan las historias de usuario que no pasaron el test de aceptación que se realizó al terminar la iteración anterior. Estas historias de usuario son divididas en tareas de entre 1 y 3 días de duración que se asignarán a los programadores. [19]
- ✓ **Reuniones diarias:** Se realizan con el objetivo de mantener una comunicación entre los desarrolladores, para así dar a conocer sus problemas y las posibles soluciones. [19]
- **Fase II. Diseño:** Se define el diseño del software.
  - ✓ **Simplicidad:** Realizar diseños sencillos para que su implementación sea más rápida para ahorrar tiempo y esfuerzos. [19]
  - ✓ **Soluciones (Spike):** "Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba "Spike", para explorar diferentes soluciones." [19]
  - ✓ **Recodificación (Refactoring):** Consiste en escribir de nuevo alguna parte del código de un programa, pero sin cambiar su funcionalidad, con el fin de hacerlo más simple y optimizar su funcionamiento, además sugiere que se recodifique cada vez que sea necesario. [20]
  - ✓ **Metáforas:** Se utiliza para explicar de una manera sencilla el propósito del proyecto. "Una buena metáfora debe ser fácil de comprender para el cliente y a su vez debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto. "[20]

- **Fase III. Codificación:** Se divide las funcionalidades del sistema, para facilitar el desarrollo, para luego ser unificadas y cumplir con lo que se ha planteado.
  - ✓ **Disponibilidad del Cliente:** Debido a que el cliente es una parte fundamental del proyecto, debe estar disponible durante el desarrollo del mismo. [20]
  - ✓ **Programación Dirigida por las Pruebas (Test-Driven Programming):** Primero se escriben los test por los cuales el sistema debe pasar; y posteriormente el desarrollo, el cual debe ser el mínimo para pasar estas pruebas.  
Las pruebas son realizadas por los desarrolladores, y son las que dirigen el desarrollo. [20]
  - ✓ **Programación en Pares:** El trabajo de programar es realizado por parejas, frente a un mismo ordenador; esto minimiza los errores y se logran mejores diseños, debido a que uno de los integrantes codifica y el otro es un poco más, esto se traduce en un producto de mejor calidad.
  - ✓ **Propiedad Colectiva del Código:** “Todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto”. [20]  
  
La pareja de programadores pueden hacer el cambio que sea necesario en el código para hacer las debidas correcciones.
  - ✓ **Ritmo Sostenido:** Con este se busca la manera de mantener un ritmo constante, pero sin sobrecargar al equipo. [20]
  
- **Fase IV. Pruebas:** Se realizan las respectivas pruebas al sistema, para así detectar posibles fallas.
  - ✓ **Pruebas Unitarias:** “Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-Driven

Programmng”). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código.” [20]

- ✓ **Detección y Corrección de Errores:** Los errores deben ser corregidos en el mismo instante en el que fueron encontrados, y posteriormente se realizan nuevas pruebas para rectificar que el error se haya resuelto. Las historias de usuario pasan por pruebas de aceptación. “Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación”. [20]
- ✓ **Pruebas de Aceptación:** “En esta prueba se evalúa el grado de calidad del software con relación a todos los aspectos relevantes para que el uso del producto se justifique.” [21] Se crean en base a las historias de usuario, se considera que una historia de usuario no está terminada si no ha pasado correctamente las pruebas de aceptación. [20]
- ✓ **Pruebas Funcionales:** “Se realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.” [22]

#### 2.4.1.3 Roles.

- **Programador.** Es fundamental para el desarrollo del software, es el responsable del código que produce para la construcción del sistema, además debe existir una buena comunicación entre los programadores. [6]
- **Cliente:** Establece lo que se debe realizar, esto es plasmado en las historias de usuario. Además, asigna la prioridad a dichas historias y decide cuáles se implementan en cada iteración. [23]
- **Encargado de pruebas (Tester):** Es el encargado de ayudar al cliente a escribir las pruebas funcionales y es responsable de las herramientas de soporte para pruebas. [23]

- **Encargado de seguimiento (Tracker):** Es el que brinda realimentación al equipo de desarrollo, revisa y calcula el progreso que ha tenido el proyecto y de acuerdo a esto hace una comparación con lo que se había estimado.[23]
- **Entrenador (Coach):** Es responsable de asegurar que se cumplan las prácticas, provee guías al equipo para que se siga el proceso correctamente, esto lo hace transmitiendo sus conocimientos y experiencia al resto del equipo. [23]
- **Consultor:** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas. [23]
- **Gestor (Big boss):** Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto. Guía al equipo para resolver un problema específico. [24]

#### 2.4.1.4 Ciclo de Vida.

“Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo. Se deben seguir los siguientes pasos.

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de

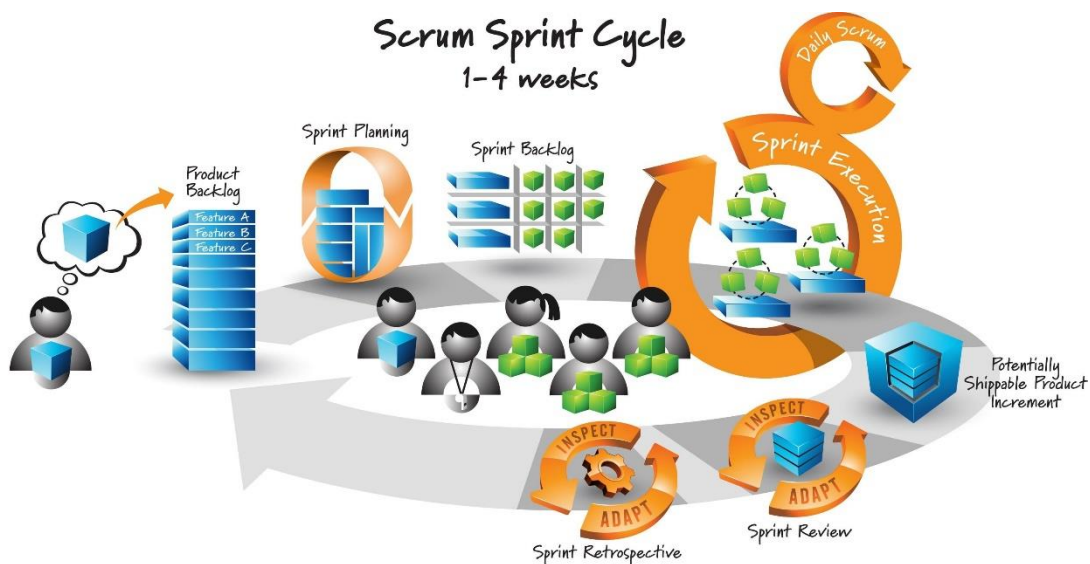
manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.” [25]

### 2.4.2 Scrum.

Es un marco de trabajo, en el que se aplican las buenas prácticas, y se pueden trabajar en equipo, para así obtener mejores resultados, “En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto [26]”.

“Scrum es una forma ágil para gestionar un proyecto, por lo general el desarrollo de software ágil con Scrum se percibe a menudo como una metodología; pero en lugar de ver Scrum como metodología, pensar en él como un marco para la gestión de un proceso”. [27]

En scrum el proyecto se realiza a través de una serie de sprints, que se entregan con un plazo máximo de 30 días, en los que debe haber un incremento funcional.



Copyright (c) 2010, Innolution, LLC & Kenneth S. Rubin. All Rights Reserved.

**Figura 5. Marco de trabajo Scrum.**

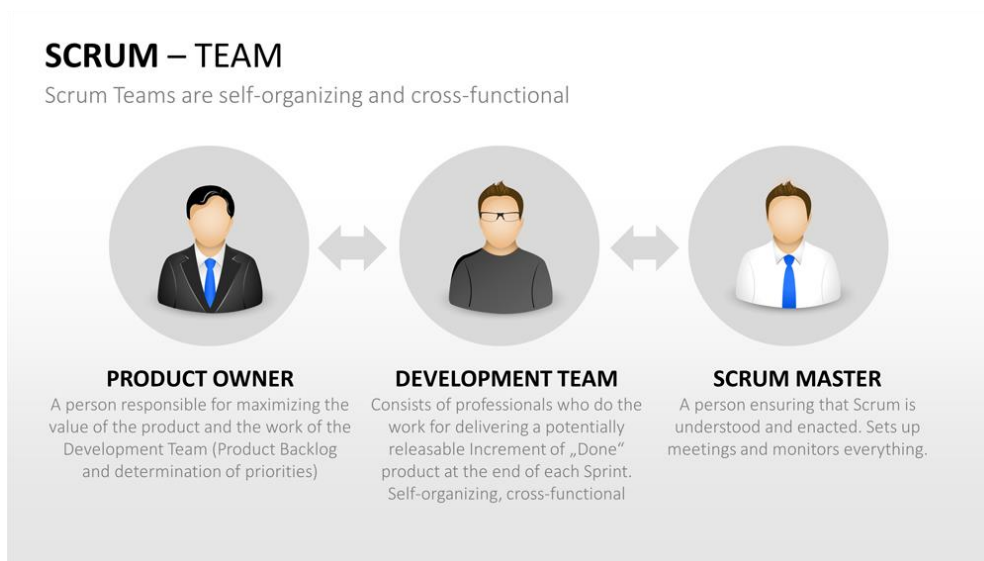
[28]

### 2.4.2.1 Roles.

#### Equipo de Scrum.

Son equipos auto-organizados, por lo tanto no son dirigidos por nadie externo, además son equipos multifuncionales, es decir, que tiene la capacidad de hacer por completo su trabajo sin depender de externos.

[29]



**Figura 6. Equipo de Scrum.**

[30]

- **Product Owner:** Es aquel que toma las decisiones, y tiene la responsabilidad de maximizar el valor del producto, además es el único capaz de gestionar la lista de productos. Se debe tener en cuenta que las decisiones del dueño de producto se reflejan en el contenido y en la priorización de la lista del producto aunque en algunos casos este puede delegar el dicha gestión al equipo de desarrollo, pero teniendo en cuenta de que la responsabilidad sigue siendo suya. [31]



- **Development Team:** Son personas que se encargan de realizar un trabajo con una asignación específica de tareas, y deben seguir un proceso para la ejecución y entrega del incremento de producto que se puede poner en producción al final de cada sprint. [29]  
Son un grupo de personas que trabajan en las historias de usuario del sprint backlog, para crear los entregables del producto, y tienen las siguientes responsabilidades:
  - ✓ Desarrollar el producto o cualquier otro resultado.
  - ✓ Comprensión de los requerimientos.
  - ✓ Hacer las estimaciones de las historias de usuarios y la creación de los entregables de producto.
  - ✓ Desempeñar el trabajo para hacer la entrega del producto.
- **Scrum Master:** Es el líder del equipo, debe asegurarse que SCRUM es entendido, adoptado, y que además se están siguiendo sus valores, reglas y procesos. [32]

Proporciona:

“Asesoría y formación al equipo para trabajar de forma auto-organizada y con responsabilidad de equipo.

- ✓ Revisión y validación de la pila del producto.
- ✓ Moderación de las reuniones.
- ✓ Resolución de impedimentos que en el sprint pueden entorpecer la ejecución de las tareas.
- ✓ Gestión de las “dinámicas de grupo” en el equipo.
- ✓ Configuración, diseño y mejora continua de las prácticas de scrum en la organización. Respeto de la organización y los implicados, con las pautas de tiempos y formas de scrum.” [29]

### 2.4.2.2 Artefactos.



**Figura 7. Artefactos de Scrum.**

[33]

- **Product Backlog:** “Lista de requisitos de usuario, que a partir de la visión inicial del producto crece y evoluciona durante el desarrollo.” [34]

Este es creado y organizado únicamente por el product owner, y nunca está completa, porque va evolucionando al mismo tiempo que el producto y el entorno en el que se usará también lo hace.

“El refinamiento de la lista de producto es el acto de añadir detalle, estimaciones y orden a los elementos de la lista de producto. Se trata de un proceso continuo, en el cual el dueño de producto y el equipo de desarrollo colaboran acerca de los detalles de los elementos de la lista de producto. Durante el refinamiento de la Lista de Producto, se examinan y revisan sus elementos.” [31]

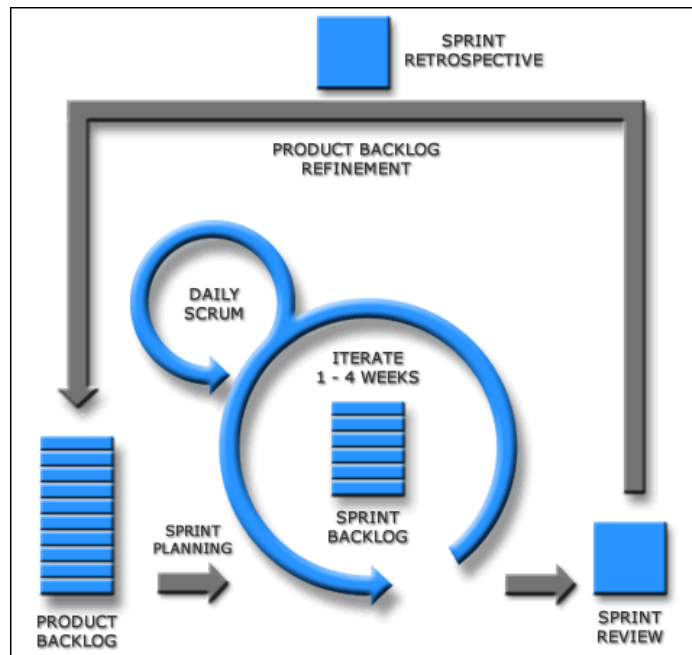
- **Sprint Backlog:** “Lista de los trabajos que debe realizar el equipo durante el sprint para generar el incremento previsto.” [34]

Dicha lista de trabajos pendientes hace ver lo que el equipo de desarrollo identifica como una necesidad para alcanzar el sprint y modifica la lista de trabajos pendientes, la cual surge a medida que se realiza el sprint, pero Solo el equipo de desarrollo puede cambiar su lista de pendientes del sprint durante un sprint.

“El seguimiento del progreso del sprint en cualquier momento durante un Sprint, es posible sumar el trabajo restante total en los elementos de la lista de pendientes del Sprint. El equipo de desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum diario para proyectar la posibilidad de conseguir el objetivo del Sprint. Haciendo seguimiento del trabajo restante a lo largo del Sprint, el Equipo de Desarrollo puede gestionar su progreso.” [31]

- **Incremento:** “Es la suma de todos los elementos de la Lista de producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar terminado” [31]

### 2.4.2.3 Eventos.



**Figura 8. Eventos de Scrum.**

[35]

- **Sprint:** “Nombre que recibe cada iteración de desarrollo. Es el núcleo central que genera el pulso de avance por tiempos prefijados (time box)” [29]
- **Agile Inception:** “Es un conjunto de dinámicas orientadas a enfocar a todas las personas involucradas en un proyecto hacia un mismo objetivo, reduciendo muchas de las incertidumbres, ayudando a explicitar los riesgos más evidentes y poniendo en común las expectativas de todos” [36], es decir que luego de dar inicio al proyecto, las personas que lo integran deben de dar claridad a las dudas que se tengan, para así poder cumplir el objetivo que se plantea.

Para realizar un inception se deben hacer las siguientes preguntas:

- ✓ ¿Por qué estamos aquí? Hacer una frase que resuma el propósito del producto.

- ✓ Elevator Pitch. Definir el proyecto en un tiempo determinado.
  - ✓ Diseña tu caja. Explicar los beneficios del producto en la caja.
  - ✓ Crea tu Not List. Describir lo que se hará, lo que se no se hará, y lo que se puede discutir para luego hacerlo.
  - ✓ ¿Qué te quita el sueño? Describir los miedos y cómo podemos minimizarlos o evitarlos.
- 
- **Planeación de la Pila de Sprint:** “La planificación de Sprint es una reunión crítica, probablemente la más importante de Scrum (en mi subjetiva opinión, por supuesto). Una planificación de Sprint mal ejecutada puede arruinar por completo todo el Sprint. El propósito de la planificación de Sprint es proporcionar al equipo suficiente información como para que puedan trabajar en paz y sin interrupciones durante unas pocas semanas, y para ofrecer al Dueño de Producto suficiente confianza como para permitirse.” [37]
  
  - **Scrum Diario (DAILY SCRUM):** “El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente. El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad.” [32]
  
  - **Sprint Review:** “El equipo presenta lo que se ha realizado en el sprint, en una forma de incremento del producto, por lo cual solo será algo funcional”. [38]

El objetivo de la reunión de Revisión del Sprint es la de inspeccionar el Incremento, si es necesario volver a analizar la Lista de Producto.

“En ésta reunión que tiene una tiene una máxima duración de 4 horas al mes, deben de participar cada uno de los que conforman los roles de scrum, es decir, el product owner, scrum team, scrum master”. [39]

Algo muy importante de la reunión, es que se evalúa si se ha cumplido lo que se propuso a alcanzar en el sprint.

- **Retrospectiva:** “Con el objetivo de mejorar de manera continua su productividad y la calidad del producto que está desarrollando, el equipo analiza cómo ha sido su manera de trabajar durante la iteración, por qué está consiguiendo o no los objetivos a que se comprometió al inicio de la iteración y por qué el incremento de producto que acaba de demostrar al cliente era lo que él esperaba o no:
  - ✓ Qué cosas han funcionado bien.
  - ✓ Cuales hay que mejorar.
  - ✓ Qué cosas quiere probar hacer en la siguiente iteración.
  - ✓ Qué ha aprendido.
  - ✓ Cuáles son los problemas que podrían impedirle progresar adecuadamente.

El Facilitador se encargará de ir eliminando los obstáculos identificados que el propio equipo no pueda resolver por sí mismo.” [40]

- **Refinamiento:** “En las reuniones de planificación de entregas y durante el transcurso de una iteración, el cliente va trabajando en

la lista de objetivos/requisitos priorizada del producto o proyecto, añadiendo requisitos, modificándolos, eliminándolos, re priorizándolos y detallando los requisitos conforme se acerca el momento de su desarrollo.

Los cambios en la lista de requisitos pueden ser debidos a:

- ✓ Modificaciones que el cliente solicita tras la demostración que el equipo realiza al final de cada iteración sobre los resultados obtenidos, ahora que el cliente entiende mejor el producto o proyecto.
- ✓ Cambios en el contexto del proyecto.
- ✓ Nuevos requisitos o tareas como resultado de nuevos riesgos en el proyecto. “ [41]

### **2.4.3 Patrón de Diseño (MVC)**

Es un patrón de diseño de software que permite separar los datos y la lógica de las interfaces del usuario mediante tres componentes modelo, vista y controlador.

#### **2.4.3.1 Modelo**

“Contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia”. [42]

#### **2.4.3.2 La Vista**

“También llamada interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste”. [42]

#### **2.4.3.4 El Controlador**

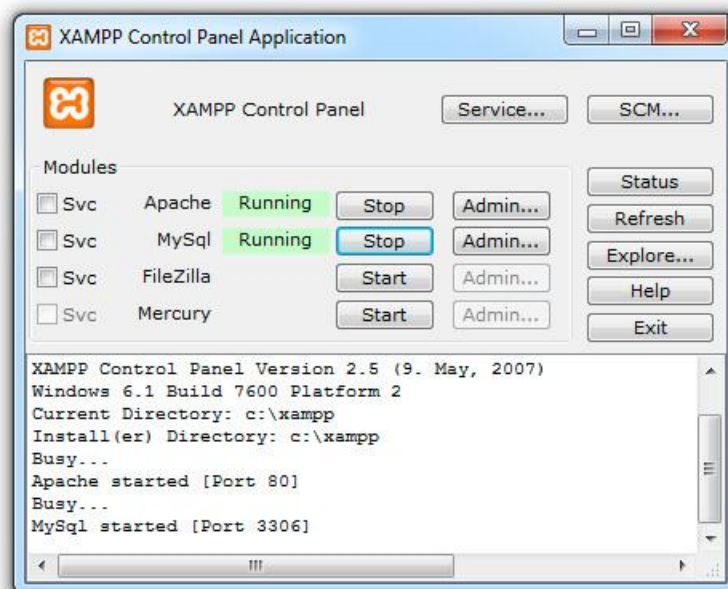
“Actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno”. [42]

## 2.5 HERRAMIENTAS DE APOYO.

### 2.5.1 Xampp.

“Es un software libre multiplataforma que incluye distribuciones de Apache, MySql, PHP y Perl, fácil de instalar y de poner en marcha en cualquier sistema operativo.

XAMPP es perfecto para aprender lenguajes como PHP o MySql sin necesidad de contratar ningún hosting, ya que desde tu mismo ordenador podrás crear y visualizar la página en la que estás trabajando.” [43]



**Figura 9. Xampp.**

[44]

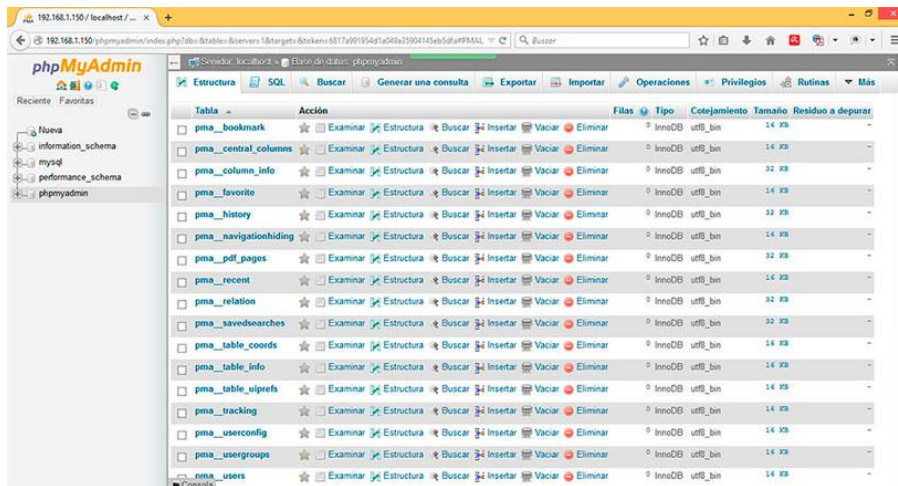
### 2.5.2 PhpMyAdmin.

“Es una herramienta de software libre escrito en PHP , la intención de manejar la administración de MySQLa través de Internet. phpMyAdmin es



compatible con una amplia gama de operaciones de MySQL y MariaDB.(Operaciones de gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, etc) se puede realizar a través de la interfaz de usuario que se utiliza con frecuencia, mientras que todavía tiene la capacidad de ejecutar directamente cualquier sentencia SQL”.

[45]

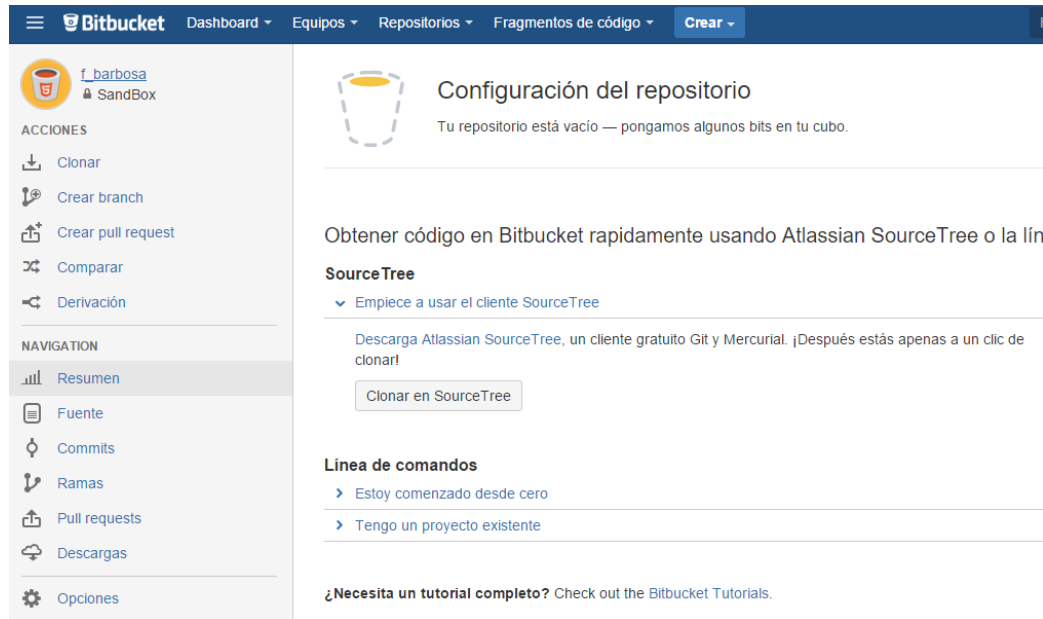


**Figura 10. PhpMyAdmin.**

[46]

### 2.5.3 Bitbucket.

“Es un servicio de alojamiento basado en la web para proyectos que utilizan el sistema de control de versiones Mercurial y Git, además ofrece cuentas gratuitas y comerciales. Las gratuitas cuentan con número ilimitado de repositorios privados y cinco usuarios, aunque tienen la opción de llegar a un total de 8 si invitas a 3 a unirse al servicio.” [47]



**Figura 11. Bitbucket.**

[48]

#### **2.5.4 Javascript.**

“Es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.” [49]

#### **2.5.5 Bootstrap.**

“Es un entorno de desarrollo con una serie de recursos que simplifican el desarrollo de un proyecto web con html5, css3 y JQuery, de manera que simplifica mucho el trabajo a la hora de diseñar, ya que el framework bootstrap ya tiene una buena parte del trabajo hecho lo cual simplifica mucho la tarea del desarrollo.” [50]

### **2.5.6 JQuery.**

“Es una biblioteca gratuita de Javascript, cuyo objetivo principal es simplificar las tareas de creación de páginas web responsivas, acordes a lo estipulado en la Web 2.0, la cual funciona en todos los navegadores modernos. Por otro lado, se dice que jQuery ayuda a que nos concentremos de gran manera en el diseño del sitio, al abstraer por completo todas las características específicas de cada uno de los navegadores.” [51]

### **2.5.7 Php Mailer.**

“PHPMailer es una clase php para enviar emails basada en el componente active server ASPMail. Permite de una forma sencilla tareas complejas como por ejemplo: Enviar mensajes de correo con ficheros adjuntos, enviar mensajes de correo en formato HTML.” [52]

### **2.5.8 Apache (SERVIDOR).**

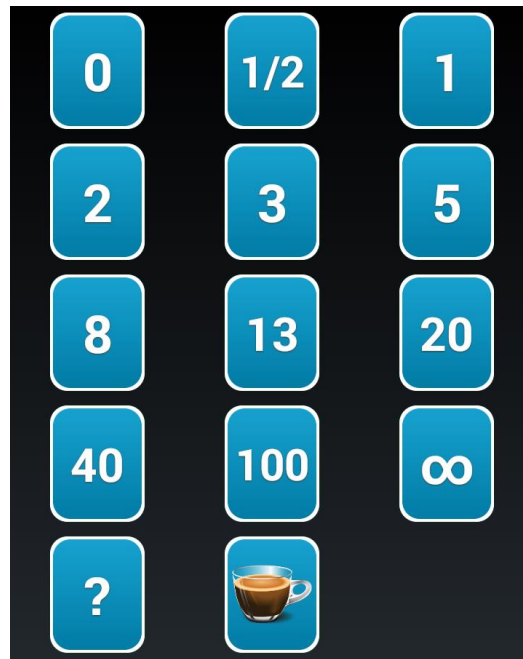
“Es un poderoso servidor web, cuyo nombre proviene de la frase inglesa “a patchy server” y es completamente libre, ya que es un software Open Source y con licencia GPL. Una de las ventajas más grandes de Apache, es que es un servidor web multiplataforma, es decir, puede trabajar con diferentes sistemas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

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.” [53]

### **2.5.9 Scrum Póker.**

“La técnica de planning poker permite hacer una estimación inicial del proyecto rápida y fiable, dado que todos los miembros del equipo comparten sus diferentes informaciones y expresan su opinión sin

sentirse condicionados por el resto. Cada número significa un peso / esfuerzo / complejidad para completar un objetivo (historia de usuario). La numeración de las cartas está basada en la sucesión de Fibonacci. La distancia entre números crece conforme se hacen mayores.” [54]

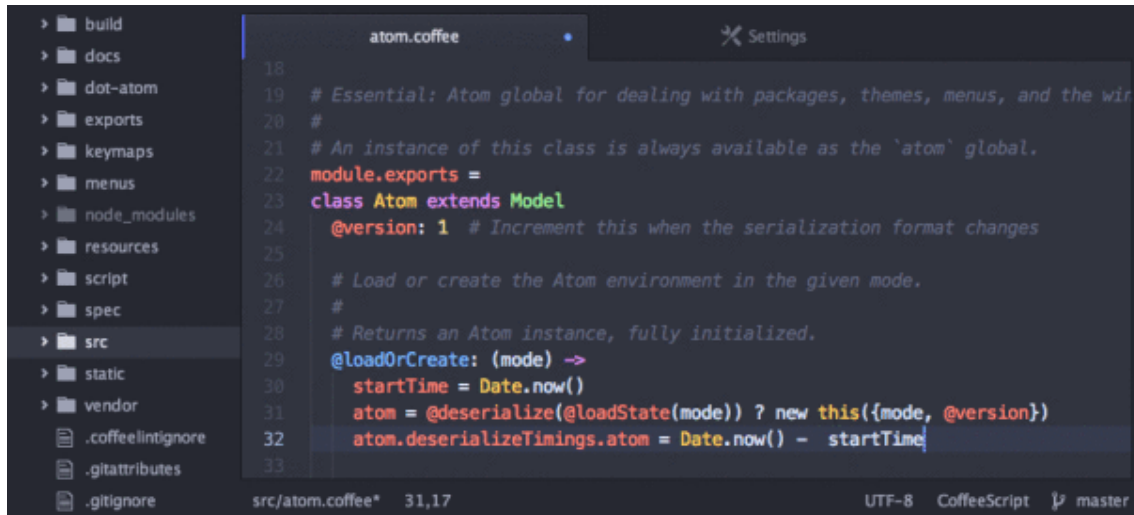


**Figura 12. Scrum Póker.**

[55]

### **2.5.10 Atom.**

“Es un editor de texto que es moderno, accesible, sin embargo hackable al núcleo-una herramienta que puede personalizar para hacer cualquier cosa, sino también utilizar de manera productiva sin tener que tocar un archivo de configuración.” [56]



```
18
19 # Essential: Atom global for dealing with packages, themes, menus, and the wir
20 #
21 # An instance of this class is always available as the `atom` global.
22 module.exports =
23   class Atom extends Model
24     @version: 1 # Increment this when the serialization format changes
25
26     # Load or create the Atom environment in the given mode.
27     #
28     # Returns an Atom instance, fully initialized.
29     @loadOrCreate: (mode) ->
30       startTime = Date.now()
31       atom = @deserialize(@loadState(mode)) ? new this({mode, @version})
32       atom.deserializeTimings.atom = Date.now() - startTime
33
```

**Figura 13. Atom.**

[57]

## 2.6 GLOSARIO

- **Plataforma web:** Conjunto de programas y sistemas que hacen que un sitio web se encuentre en funcionamiento.
- **Gestión:** “Conjunto de trámites que se llevan a cabo para resolver un asunto.” [58]
- **Simplicidad:** Realizar diseños sencillo y amigables con el usuario, disminuyendo tiempo y costos.
- **Retroalimentación:** “es un proceso por el que una cierta proporción de la señal de salida de un sistema se redirige de nuevo a la entrada. Esto es frecuentemente en el control del comportamiento dinámico del sistema.” [59]
- **Planificación:** Es una fase de la metodología ágil extreme programming, en la cual se determina el objetivo del proyecto.
- **Test de aceptación:** “Es una prueba, que intenta obtener ciertos resultados comprobatorios.” [60]
- **Iteración:** “Es un conjunto de periodos de tiempo dentro de un proyecto”. [61]

- **Codificación:** Es una fase de la metodología ágil extreme programming, Se divide las funcionalidades del sistema, para facilitar el desarrollo.
- **Prueba unitaria:** Es una fase de la metodología ágil extreme programming, en la cual se detectan fallas.
- **Framework:** “Es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, librerías y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.” [62]
- **Metodología:** Grupo de mecanismos o procedimientos racionales, empleados para el logro de un objetivo. [63]
- **Scrum:** Es un marco de trabajo, en el que se aplican las buenas prácticas, las cuales se pueden trabajar en equipo, para así obtener mejores resultados
- **Diseño:** Es una fase de la metodología ágil extreme programming, en la cual se define la arquitectura del sistema.
- **Product owner:** Es aquel que se encarga de priorizar mediante el product backlog, lo que se realizara en los sprints.
- **Development team:** Son los encargados de desarrollar e implementar el software, y son auto-organizados.
- **Scrum master:** Es el encargado de apoyar al development team y al product owner, durante el desarrollo del proyecto.
- **Dayli:** Son reuniones que se hacen a diario para, ver el progreso de los sprints.

### 3. METODOLOGÍA.

Para la implementación de la plataforma web se hizo uso de la metodología ágil XP, debido a que el cliente tiene más participación e interacción con el equipo de desarrollo, además nos facilitó el desarrollo de la plataforma debido a que se usa programación en parejas, permitiendo así hacer una retroalimentación continua.

Para la gestión del proyecto se hizo uso de las prácticas de Scrum, ya que éste promueve el trabajo en equipo, exigiendo entregas parciales funcionales del desarrollo en cada uno de los sprints.

#### 3.1 MARCO DE TRABAJO SCRUM.

En el Sprint Planning se decidió que se realizarían una serie de SPRINT, los cuales se presentará a continuación:

##### 3.1.1 Planificación del Sprint 0.

###### 3.1.1.1 Inception.

Lo primero que se realizó fue una asignación de roles del Scrum Team, para así enfocar a cada uno de los integrantes hacia el mismo objetivo y dar claridad a las incertidumbres, para cumplir el objetivo.

**Tabla 1. Scrum Master**

<b>Nombre</b>	Julián Darío Bermúdez Trujillo
<b>Rol</b>	Scrum Master
<b>Responsabilidad</b>	Facilitar y acompañar el desarrollo del producto.

**Tabla 2. Development Team**

<b>Nombre</b>	Octavio Andrés Cifuentes Dorado
<b>Rol</b>	Investigador y Desarrollador
<b>Responsabilidad</b>	Investigar y Desarrollar el producto.

**Tabla 3. Development Team**

<b>Nombre</b>	Andrés Felipe Vásquez
<b>Rol</b>	Investigador y Desarrollador
<b>Responsabilidad</b>	Investigar y Desarrollar el producto.

**Tabla 4. Product Owner**

<b>Nombre</b>	Santiago Sánchez Ferreira
<b>Rol</b>	Product Owner
<b>Responsabilidad</b>	Representar al comité de investigaciones del programa de Ingeniería de sistemas informáticos.



- **Product Backlog:** Se realizó una lista priorizada de las actividades a implementar, esto se hizo con el fin de hacer primero aquellas funcionalidades con mayor valor, en esta lista se indican las entregas que espera el cliente, además de hacer un costo estimado para la viabilidad del proyecto.

Para este proceso se basó en las prácticas de agile inception que se mencionan a continuación:

- ✓ ¿Por qué estamos aquí?
- ✓ Elevator Pitch.
- ✓ Diseña tu caja.
- ✓ Crea tu Not List.
- ✓ ¿Qué te quita el sueño?

### **3.1.2 Planificación del Sprint 1.**

Se encuentra conformado por 7 historias de usuario, las cuales se estimaron con un tiempo de ejecución de 18 días (hábiles) de 8 horas de trabajo cada uno.

Estas son:

- Autenticar Usuario.
- Registrar Usuario.
- Cargar Archivo.
- Consultar Información Estudiante.
- Cambiar Contraseña.
- Consultar Reportes.
- Consultar Información Miembro Comité.

### **3.1.3 Planificación del Sprint 2.**

Se encuentra conformado por 6 historias de usuario, las cuales se estimaron con un tiempo de ejecución de 16 días (hábiles) de 8 horas de trabajo cada uno.

- Consultar Información Jurado.
- Recibir Alerta.
- Consultar Información Docente.
- Rechazar Proyecto
- Asignar Jurados.
- Asignar Fecha de Sustentación.

#### **3.1.4 Planificación del Sprint 3.**

Se encuentra conformado por 7 historias de usuario, las cuales se estimaron con un tiempo de ejecución de 16 días (hábiles) de 8 horas de trabajo cada uno.

- Crear Jurado.
- Recuperar Contraseña.
- Crear Anteproyecto.
- Crear Proyecto.
- Asignar Concepto.
- Crear Director Proyecto.
- Crear Revisión.

#### **3.1.5 Planificación Daily Scrum.**

Durante el desarrollo del proyecto se realizaron reuniones siempre en el mismo lugar con una duración de 15 minutos, con el fin de dar a conocer lo que se hizo, los problemas que se tuvieron para realizar la actividad, y así planear lo que se va a hacer en el siguiente sprint. Estas reuniones nos permitieron hacer una retroalimentación continua en cada sprint.

Durante las primeras dos semanas de trabajo ésta reunión la facilita el scrum master, posteriormente se delega directamente al equipo de desarrollo y el scrum master participa de alguna de ellas.

### 3.1.6 Sprint Review.

El equipo de desarrollo con el Scrum Master, comparan los resultados con lo esperado del producto (incremento).

### 3.1.7 Retrospectiva.

Entre el equipo de desarrollo y el scrum master, para evaluar y mejorar no solamente los aspectos del proceso del producto, sino el proceso de la realización del trabajo de grado.

Se hicieron mediante el método de la estrella de mar que se basa en “cinco áreas específicas para tratar, evitando así centrarse solo en lo bueno o en lo malo.”[64] Estas son: Comenzar a hacer, más de, seguir haciendo, menos de, dejar de hacer.



**Figura 14. Retrospectiva Estrella de Mar.**

[64]

Esta técnica se implementó haciendo reuniones entre el Scrum master, development team y el product owner.

### **3.2 XP (XTREME PROGRAMMING).**

Para conocer las necesidades que realmente tenían en la Corporación Universitaria Autónoma del Cauca, para gestionar la información y hacer seguimiento a los trabajos de grado de los estudiantes; se hicieron una serie de entrevistas, encuestas y reuniones con los miembros interesados en este proceso, obteniendo información valiosa para darle solución a este problema.

#### **3.2.1 Planificación.**

Para determinar los procesos que actualmente maneja la facultad de ingeniería para la gestión de la información, se realizó una serie de entrevistas y encuestas a los involucrados, las cuales permitieron mediante las historias de usuario tener más detallado lo que los usuarios requería. A dichas historias se les dió una estimación del tamaño y del peso con algunos valores como son: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, los cuales fueron extraídos y usados de la herramienta scrum póker.

La metodología de XP, sugiere hacer reuniones diarias, que se realizaron de manera presencial, para así ver el progreso y las fallas presentadas durante el desarrollo del proyecto, permitiendo aclarar inquietudes y brindar posibles soluciones a problemas propiciar una retroalimentación continua durante todo el proceso.

#### **3.2.2 Diseño.**

En esta fase, se tuvo en cuenta el diseño sencillo, no solo para darle buena presentación y facilidad de uso, sino también para ahorrar tiempo durante el desarrollo.

#### **3.2.3 Codificación.**

Para el desarrollo de la plataforma web, se usó el lenguaje de programación Php porque es libre, permite un fácil acceso a las bases de

datos y además es un lenguaje creado para el desarrollo web que se ejecuta por el lado del servidor, MySQL como gestor de base de datos y el framework Yii2 debido a su escalabilidad, estas dos últimas herramientas utilizan el modelo, vista, controlador (MVC).

Se usó la herramienta Xampp, para de esta misma utilizar el Apache, MySQL y PhpMyAdmin para administrar la base de datos.

Por adaptabilidad y una interfaz que puede ser adaptativa para el agrado del usuario final, se manejó Bootstrap, y dentro de esta herramienta se hizo uso de HTML como lenguaje de maquetado y CSS para la presentación y estilo de la plataforma web.

Por otro lado para el envío de alertas, se hizo uso de PhpMailer, ya que estas se harán por medio de los correos.

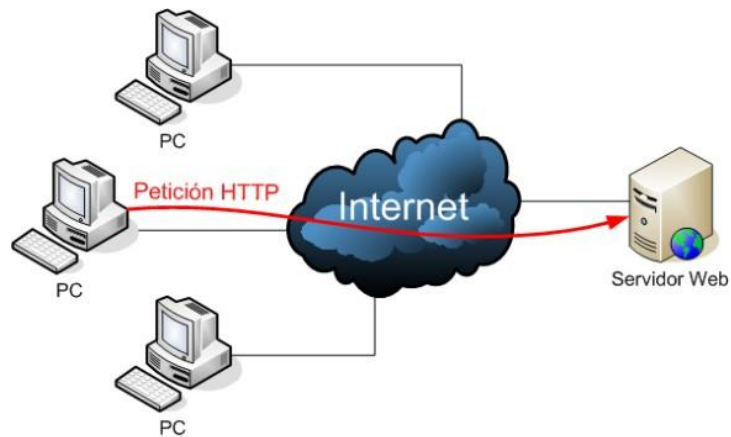
Para la fácil codificación e implementación del lenguaje de programación se utilizó el editor de texto Atom como IDLE, debido a que es gratis y posee una agradable interfaz que facilita su uso.

#### **3.2.4 Pruebas.**

Se hicieron pruebas unitarias, es decir que a cada uno de los módulos del código se realizaron este tipo de pruebas. También se hicieron pruebas de aceptación a cada una de las historias de usuario, para así dar por bien hecha cada una de ellas basándose en el entorno de trabajo Scrum, debido a que en este se requieren hacer entregas constantes y funcionales. Y por último se realizaron pruebas funcionales a la interfaz de la plataforma web, para conocer su funcionalidad.

### **3.3 ARQUITECTURA DEL SISTEMA.**

Se hizo uso de la arquitectura cliente-servidor, debido a que uno o más usuarios pueden obtener la información solicitada en el momento en que se desee, es decir que soporta concurrencia. Además de brindar seguridad a la plataforma web.



**Figura 15. Arquitectura Cliente-Servidor.**

[65]

### **3.3.1 Cliente**

En este tipo de arquitectura el cliente es el que realiza las peticiones al servidor.

Durante el desarrollo de la plataforma web, los clientes fueron los usuarios involucrados en el proceso de gestión de información y seguimiento de trabajos de grado de la facultad de ingeniería, los cuales pueden ingresar a la plataforma desde cualquier dispositivo con acceso a internet.

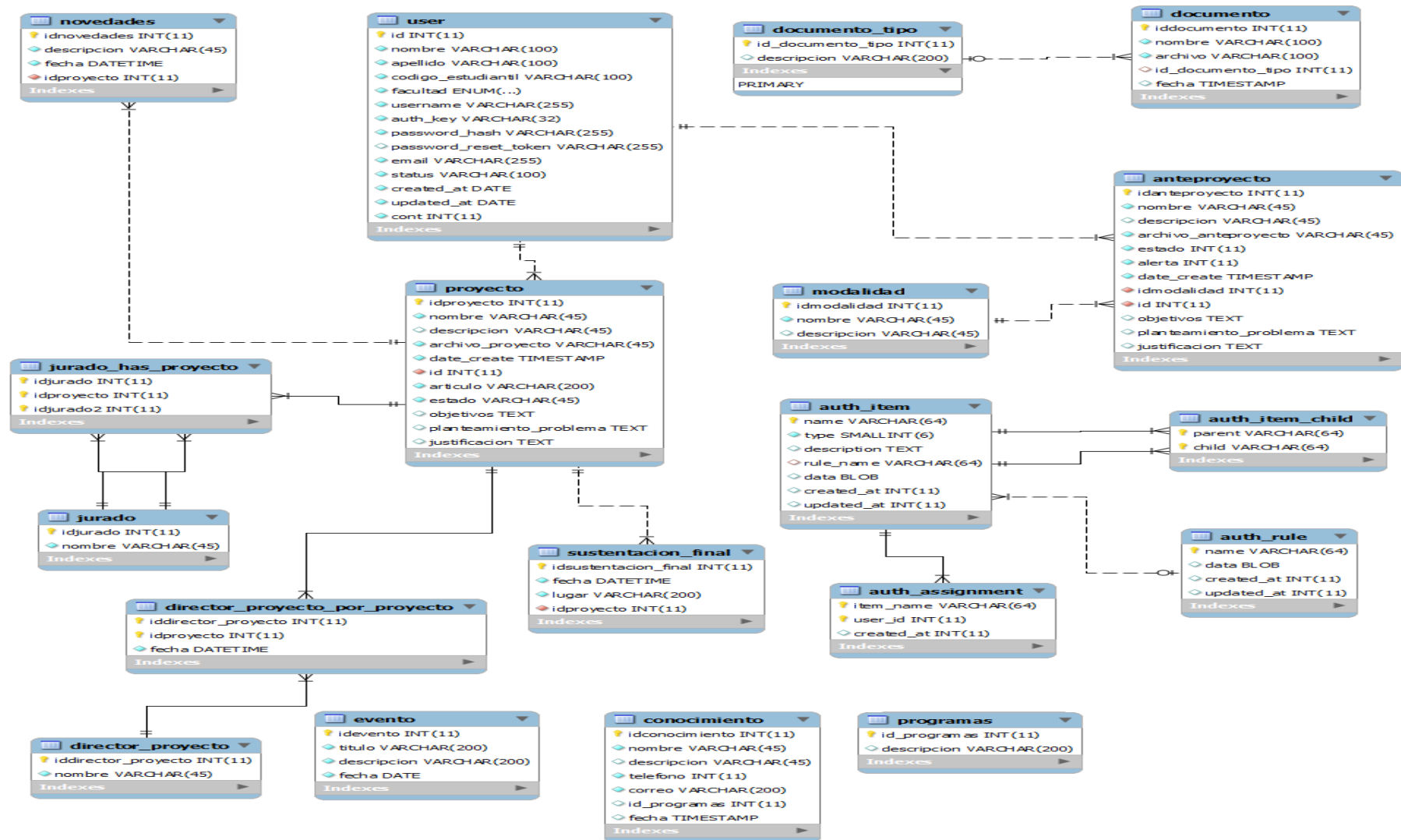
### **3.3.2 Servidor**

En este tipo de arquitectura el servidor es el que luego de recibir una solicitud, procesa el código de la plataforma (Php, HTML, CSS, MySql) y posteriormente brinda una respuesta a los clientes.

## **4. INGENIERÍA**

### **4.1 IMPLEMENTACIÓN DEL SPRINT 0.**

#### **4.1.2 Diagrama de Base de Datos**





**Figura 16. Diagrama de Base de Datos. [Fuente Propia]**

#### **4.1.3 Desarrollo de Agile Inception.**

Se realizó la implementación de ésta técnica, fue una parte fundamental para el buen desarrollo del proyecto, debido a que permitió tener un poco más claro la necesidad del Product Owner, y así cumplir el objetivo y definir el producto final esperado, dando respuesta a las siguientes preguntas.

- **¿Por qué estamos aquí?**

Estamos aquí para facilitar la gestión de la información y seguimiento de los trabajos de grado en la facultad de ingeniería de la Corporación Universitaria Autónoma del Cauca, esto a través de una plataforma web.

- ✓ **¿Por qué estamos haciendo este producto?**

**Razón 1:** Automatizar el proceso de trabajos de grado.

**Razón 2:** Disminución de documentación física.

**Razón Principal:** Disminución de tiempo y costo.

- **Elevator Pitch.**

Es un proyecto para la facultad de ingeniería de la Corporación Universitaria Autónoma del Cauca que necesita gestionar la información y hacer seguimiento de los trabajos de grado a su cargo, este producto se llama SGTG (Sistema de Gestión de trabajos de grado), y es una plataforma web cuya ventaja frente al proceso actual es que agiliza los tiempos de respuesta de las personas involucradas en este.

- **Diseña tu caja.**



**Figura 17. Diseña tu caja [Fuente Propia]**

- **Crea tu Not List.**

- ✓ **IN**

- Gestionar Usuarios.
- Guardar Historial.
- Banco de Proyectos.
- Generar Reportes.
- Tener seguridad de la información.

- ✓ **OUT**

- Manejar información de otras facultades.

- ✓ **DISCUTIR**

- Chat interno entre los involucrados.

- **¿Qué te quita el sueño?**

- ✓ **¿Cuáles son tus miedos?**

- Que el sistema deje de funcionar por problemas de conectividad.

- Que se dificulte la curva de aprendizaje de los usuarios.
- ✓ **¿Cómo evitarlos?**
  - Implementándolo en un buen servicio de internet o contando con el apoyo de la infraestructura y equipo de TI de la institución.
  - Realizando un buen proceso de capacitación.
  - Dejando buena documentación.
- ✓ **¿Cómo minimizar el riesgo?**
  - Siguiendo el plan para el desarrollo.
  - Brindando a los usuarios finales lo que necesitan.
- ✓ **¿Qué nos mantiene despiertos por las noches?**
  - **Riesgo 1:** Los usuarios no queden a gusto con la plataforma.
  - **Riesgo 2:** El desarrollo no se termine en el tiempo establecido.
  - **Impedimento 1:** La interfaz no sea de buen agrado para los usuarios.
  - **Impedimento 2:** Existan nuevas funcionalidades.

#### **4.1.3.1 Sprint Review**

Para hacer un estimación de tiempo, recursos y cumplir en la entrega de cada uno de los sprint, se hizo la respectiva priorización de las historias de usuario. Se hizo una socialización de las funcionalidades que propiciaría el producto final.

#### **4.1.3.2 Retrospectiva**

Se hizo uso de la técnica de la estrella de mar, ya que permite tener retroalimentación, y acorde a esto se ajusta la ejecución de actividades para el siguiente sprint.

**Tabla 5. Retrospectiva del Sprint 0.**

<p><b>Comenzar a Hacer.</b></p> <ul style="list-style-type: none"> <li>• Documentación del proceso de desarrollo.</li> <li>• Reuniones con el cliente.</li> </ul>	<p><b>Más de.</b></p> <ul style="list-style-type: none"> <li>• Reuniones del equipo.</li> <li>• Equipo comprometido.</li> </ul>
<p><b>Seguir Haciendo.</b></p> <ul style="list-style-type: none"> <li>• Compartir la documentación encontrada con el equipo de trabajo.</li> </ul>	<p><b>Menos de.</b></p> <ul style="list-style-type: none"> <li>• Llegar tarde a las reuniones del equipo.</li> </ul>
<p><b>Dejar de Hacer.</b></p> <ul style="list-style-type: none"> <li>• Conversaciones que no sean relacionadas al desarrollo del proyecto que distraigan al equipo.</li> </ul>	

## 4.2 IMPLEMENTACIÓN DEL SPRINT 1

Para desarrollar 7 historias de usuario y así dar finalización a la primera entrega o iteración, en un tiempo de ejecución de 18 días (hábiles) de 8 horas de trabajo cada uno, se realizó planeación, diseño, desarrollo y pruebas.

Estas son:

- Autenticar Usuario.
- Registrar Usuario.
- Seleccionar Archivo.
- Consultar Información Estudiante.
- Cambiar Contraseña.
- Consultar Reportes.
- Consultar Información Miembro Comité.

### 4.2.1 Planificación.

A continuación se muestran las historias de usuario y task card del Sprint 1.

**Tabla 6. Historia de Usuario Autenticar Usuario.**

<b>Historia de usuario</b>	
<b>Número:</b> HU001	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Autenticar Usuario	
<b>Nivel de riesgo de desarrollo:</b> medio	
<b>Peso de la historia:</b> 1 punto de historia	
<b>Tiempo estimado:</b> 11 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Requiero:</b> autenticarme en la plataforma	
<b>Para:</b> Poder acceder a las funciones correspondientes al rol.	
<b>Criterios de aceptación</b>	
<ol style="list-style-type: none"> <li>1. Registro exitoso: Cuando se encuentre habilitado por el administrador y se ingresen los datos correctamente (usuario y contraseña como campos obligatorios) se permitirá el acceso a la plataforma.</li> <li>2. Registro no exitoso: cuando se ingresen los datos de manera incorrecta se mostrará un mensaje de error y no se permitirá el acceso a la plataforma.</li> </ol>	
<b>Restricciones:</b>	
<ol style="list-style-type: none"> <li>1. El usuario asignado debe ser único</li> </ol>	
<b>Actividades</b>	<b>N° de horas de implementación</b>

<ul style="list-style-type: none"> <li>• Crear Interfaces.</li> </ul>	2
<ul style="list-style-type: none"> <li>• Conexión al servidor.</li> </ul>	3
<ul style="list-style-type: none"> <li>• Validación de la información ingresada por el usuario por medio de la interfaz.</li> </ul>	1
<ul style="list-style-type: none"> <li>• Testing</li> </ul>	2
<ul style="list-style-type: none"> <li>• Integración</li> </ul>	3

**Tabla 7. Task Card Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU001:</b> Autenticar Usuario
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 07/03/2017	<b>Fecha Fin:</b> 7/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para autenticar usuario: usuario, contraseña e iniciar sesión	

**Tabla 8. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU001:</b> Autenticar Usuario
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 08/03/2017	<b>Fecha Fin:</b> 08/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso autenticar usuario.	

**Tabla 9. Task Card: Validación de la información ingresada por el usuario por medio de la interfaz.**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU001:</b> Autenticar Usuario
<b>Nombre de Tarea:</b> Validación de la información ingresada por el usuario por medio de la interfaz	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 1 Horas	

<b>Fecha Inicio:</b> 08/03/2017	<b>Fecha Fin:</b> 08/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, etc)	

**Tabla 10. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU001:</b> Autenticar Usuario
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 08/03/2017	<b>Fecha Fin:</b> 08/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	



**Tabla 11. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU001:</b> Autenticar Usuario
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 08/03/2017	<b>Fecha Fin:</b> 09/03/2017
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).</p>	

**Tabla 12. Historia de usuario Registrar usuario**

<b>Historia de usuario</b>
<b>Número:</b> HU002
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Registrar Usuario
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 1 punto de historia
<b>Tiempo estimado:</b> 17 Horas

### Perspectiva del producto

**Como:** Usuario Administrador

**Requiero:** Registrar un nuevo usuario

**Para:** Permitirle acceso a la funcionalidad de la plataforma.

### Criterios de aceptación

1. Registro de un nuevo usuario exitoso: Llenar campos obligatorios: nombre, apellido, identificación, programa, usuario, email, rol, nombre de usuario y contraseña; la plataforma informará que el registro se realizó satisfactoriamente
2. Registro fallido: en caso de no contar con los campos requeridos la plataforma informará qué campos hacen falta para poder crear el registro
3. El Nombre de usuario y la contraseña serán asignados por el administrador.

### Restricciones

- Los usuarios solo podrán ser creados por el usuario administrador.
- El administrador proporciona una contraseña inicial que deberá ser cambiada por el nuevo usuario registrado en el primer inicio de sesión.

Actividades	N° de horas de implementación
• Crear Interfaces.	2
• Conexión al servidor.	3
• Validación de datos ingresados por el usuario de acuerdo al tipo de dato (int, float, string, etc.).	3
• Crear nuevo registro con los datos ingresados por el usuario por medio de la interfaz	4
• Testing	2
• Integración	3

**Tabla 13. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU002:</b> Registrar Usuario
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 09/03/2017	<b>Fecha Fin:</b> 09/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para registrar usuario estudiante con los campos requeridos y los obligatorios serán demarcados con *.	

**Tabla 14. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU002:</b> Registrar Usuario
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	

<b>Fecha Inicio:</b> 09/03/2017	<b>Fecha Fin:</b> 09/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso registrar usuario.	

**Tabla 15. Task Card: Validación de datos ingresados por el usuario**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU002:</b> Registrar Usuario
<b>Nombre de Tarea:</b> Validación de datos ingresados por el usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 09/03/2017	<b>Fecha Fin:</b> 10/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Validar cada uno de los campos de la interfaz gráfica para el almacenamiento correcto de la información solicitada, teniendo en cuenta el tipo de dato (int, string, double, float, etc)	

**Tabla 16. Task Card: Crear nuevo registro con los datos ingresados por el usuario por medio de la interfaz**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU002:</b> Registrar Usuario
<b>Nombre de Tarea:</b> Crear nuevo registro con los datos ingresados por el usuario por medio de la interfaz.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 10/03/2017	<b>Fecha Fin:</b> 10/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear nuevo registro con los datos ingresados por el usuario por medio de la interfaz y almacenarlos en la base de datos.	

**Tabla 17. Task card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU002:</b> Registrar Usuario
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	

<b>Fecha Inicio:</b> 10/03/2017	<b>Fecha Fin:</b> 10/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 18. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 6	<b>HU002:</b> Registrar Usuario
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 1 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 10/03/2017	<b>Fecha Fin:</b> 13/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 19. Historia de usuario Seleccionar Archivo**

<b>Historia de usuario</b>
<b>Número:</b> HU003
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Seleccionar Archivo
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 3 puntos de historia
<b>Tiempo estimado:</b> 20 Horas
<b>Perspectiva del producto</b>
<p><b>Como:</b> Usuario</p> <p><b>Requiero:</b> Seleccionar un documento (informes, notificaciones, cartas) para subirlo a la plataforma.</p> <p><b>Para:</b> Enviarlo a quien corresponde.</p>
<b>Criterios de aceptación</b>
<ol style="list-style-type: none"> <li>1. Seleccionar archivo exitoso: cuando se solicita cargar archivo en los formatos especificados (.txt,.doc,.docx,) la plataforma verifica el tamaño del documento (máximo 10 megas) y permite cargarlo, informando al usuario que la acción se realizó satisfactoriamente.</li> <li>2. Seleccionar archivo con formato equivocado: si el usuario intenta cargar un archivo que no corresponda a los especificados, no se podrá cargar el archivo y le informará al usuario para que realice los cambios correspondientes.</li> <li>3. Cargar archivo de mayor tamaño: si el usuario intenta cargar un archivo de mayor tamaño al soportado por la plataforma se le informará al usuario que ha sobrepasado el tamaño.</li> </ol>

<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor.</li> <li>• Verificación de la extensión y el tamaño del documento para poder ser seleccionado.</li> <li>• Crear Seleccionar Archivo en la plataforma.</li> <li>• Testing</li> <li>• Integración</li> </ul>	<p>2</p> <p>4</p> <p>4</p> <p>4</p> <p>2</p> <p>4</p>

**Tabla 20. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU003:</b> Seleccionar Archivo
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 13/03/2017	<b>Fecha Fin:</b> 13/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para seleccionar archivo: creación de botones y etiquetas que permitan la búsqueda y selección de un archivo.	



**Tabla 21. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU003:</b> Seleccionar archivo
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 13/03/2017	<b>Fecha Fin:</b> 13/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso seleccionar archivo.	

**Tabla 22. Task card: Verificación de la extensión y tamaño del documento para poder ser seleccionado**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU003:</b> Seleccionar archivo
<ul style="list-style-type: none"> <li><b>Nombre de Tarea:</b> Verificación de la extensión del documento para poder ser cargado.</li> </ul>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	

<b>Fecha Inicio:</b> 14/03/2017	<b>Fecha Fin:</b> 14/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita comparar la extensión y tamaño del documento con los máximos permitidos en los requisitos.	

**Tabla 23. Task Card: Crear seleccionar archivo en la plataforma**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU003:</b> Seleccionar archivo
<b>Nombre de Tarea:</b> Crear seleccionar archivo en la plataforma	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 14/03/2017	<b>Fecha Fin:</b> 14/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Agregar la funcionalidad al botón seleccionar archivo para permitir al usuario elegir el archivo que desea cargar a la plataforma	

**Tabla 24. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU003:</b> Seleccionar archivo
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 15/03/2017	<b>Fecha Fin:</b> 15/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 25. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 6	<b>HU003:</b> Seleccionar archivo
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 15/03/2017	<b>Fecha Fin:</b> 15/03/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).

**Tabla 26. Historia de usuario Consultar información Estudiante**

<b>Historia de usuario</b>
<b>Número:</b> HU004
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Consultar Información Estudiante
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 5 puntos de historia
<b>Tiempo estimado:</b> 30 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario Estudiante <b>Requiero:</b> Consultar información correspondiente a: estado de anteproyecto, estado de proyecto, cronograma de comités, jurados, director, fecha de sustentación, banco de proyectos, tiempo disponible para entrega, y correcciones del proyecto o anteproyecto. <b>Para:</b> poder hacer seguimiento del proyecto en curso.
<b>Criterios de aceptación</b>
1. Consultar información: requiere que el usuario estudiante esté autenticado,

y tenga un proyecto o anteproyecto creado, una vez validada la información el usuario podrá visualizar toda la información que necesite saber, como: estado de anteproyecto o proyecto, cronograma de comités, jurados, director, fecha de sustentación, banco de proyectos, tiempo disponible para entrega, y correcciones del proyecto o anteproyecto.

2. Consultar información exitoso: en caso de que se pueda realizar la operación la plataforma desplegará la información que el usuario solicitó.
3. Consultar información no exitoso: en caso de que no exista la información disponible la plataforma informará al usuario que no se puede mostrar la información.
4. El usuario Estudiante no necesita estar autenticado en la plataforma para acceder al banco de proyectos.

Actividades	N° de horas de implementación
<ul style="list-style-type: none"> <li>• Crear interfaces</li> </ul>	5
<ul style="list-style-type: none"> <li>• Conexión al servidor</li> </ul>	5
<ul style="list-style-type: none"> <li>• Consultar información</li> </ul>	5
<ul style="list-style-type: none"> <li>• Testing</li> </ul>	8
<ul style="list-style-type: none"> <li>• Integración</li> </ul>	7

**Tabla 27. Task Card: Crear Interfaces**

TAREA DE INGENIERÍA	
<b>Número de Tarea:</b> 1	<b>HU004:</b> Consultar Información Estudiante
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia

<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 15/03/2017	<b>Fecha Fin:</b> 16/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaces gráficas para visualizar información en la plataforma: creación de botones, etiquetas y campos de texto que permitan la consulta y visualización de la información.	

**Tabla 28. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU004:</b> Consultar Información Estudiante
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 16/03/2017	<b>Fecha Fin:</b> 16/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso consultar información estudiante.	

**Tabla 29. Task Card: Consultar información**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU004:</b> Consultar Información Estudiante
<b>Nombre de Tarea:</b> Consultar información	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 17/03/2017	<b>Fecha Fin:</b> 17/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear los métodos que permitan la visualización de la información consultada por el estudiante.	

**Tabla 30. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU004:</b> Consultar Información Estudiante
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 8 Horas	
<b>Fecha Inicio:</b> 17/03/2017	<b>Fecha Fin:</b> 20/03/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.

**Tabla 31. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU004:</b> Consultar Información Estudiante
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 7 Horas	
<b>Fecha Inicio:</b> 20/03/2017	<b>Fecha Fin:</b> 21/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	



**Tabla 32. Historia de usuario Cambiar Contraseña**

<b>Historia de usuario</b>	
<b>Número:</b> HU005	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Cambiar Contraseña	
<b>Nivel de riesgo de desarrollo:</b> media	
<b>Peso de la historia:</b> 2 puntos de historia	
<b>Tiempo estimado:</b> 16 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario	
<b>Requiero:</b> Cambiar la contraseña asignada.	
<b>Para:</b> Poder tener acceso a la funcionalidad de la plataforma.	
<b>Criterios de aceptación</b>	
<ol style="list-style-type: none"> <li>1. Cambiar Contraseña exitoso: se le indica al usuario actualice la contraseña asignada por el usuario administrador, la primera vez que ingresa a la plataforma para poder continuar haciendo uso de esta. Si los datos que ingresa están correctamente validados se permitirá realizar la modificación y se informará al usuario que la acción se realizó satisfactoriamente.</li> <li>2. Gestionar perfil no exitoso: en caso de que los datos ingresados por el usuario no sean compatibles con los datos que se esperan recibir la plataforma mostrará un mensaje de error informando al usuario.</li> </ol>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> </ul>	2

<ul style="list-style-type: none"> <li>• Conexión al servidor</li> </ul>	3
<ul style="list-style-type: none"> <li>• Actualización de registro solicitado por el usuario</li> </ul>	5
<ul style="list-style-type: none"> <li>• Testing</li> </ul>	2
<ul style="list-style-type: none"> <li>• Integración</li> </ul>	4

**Tabla 33. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU005:</b> Cambiar contraseña
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 21/03/2017	<b>Fecha Fin:</b> 22/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para cambiar contraseña: crear etiquetas, campos de texto y botones (contraseña actual, contraseña nueva, repetir contraseña).	

**Tabla 34. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU005:</b> Cambiar contraseña
<b>Nombre de Tarea:</b> Conexión al servidor	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 22/03/2017	<b>Fecha Fin:</b> 22/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso cambiar contraseña.	

**Tabla 35. Task Card: Actualización de registro solicitado por el usuario**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU005:</b> Cambiar contraseña
<b>Nombre de Tarea:</b> Actualización de registro solicitado por el usuario	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 22/03/2017	<b>Fecha Fin:</b> 23/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita consultar y modificar en la base de datos la contraseña del usuario .	

**Tabla 36. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU005:</b> Cambiar contraseña
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 23/03/2017	<b>Fecha Fin:</b> 23/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 37. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU005:</b> Cambiar contraseña
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 23/03/2017	<b>Fecha Fin:</b> 23/03/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).

**Tabla 38. Historia de usuario: Consultar Reportes**

<b>Historia de usuario</b>
<b>Número:</b> HU006
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Consultar Reportes
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 5 puntos de historia
<b>Tiempo estimado:</b> 31 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario Administrador <b>Requiero:</b> Consultar reportes de los proyectos y anteproyectos en la plataforma. <b>Para:</b> Obtener la información correspondiente a directores, proyectos, estudiantes, estado de proyecto; por fechas.
<b>Criterios de aceptación</b>
1. Consultar historial exitoso: el usuario debe estar autenticado, en caso de que solicite la consulta del historial y que en la plataforma se encuentre registrado por lo menos un anteproyecto o proyecto de grado, se mostrará

<p>un archivo pdf con una tabla conteniendo toda la información existente.</p> <p>2. Consultar historial no exitoso: en caso de que no se encuentren anteproyectos o proyectos registrados en la plataforma se informará que no hay datos en el historial para poder mostrar.</p>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear Interfaces</li> <li>• Conexión al servidor</li> <li>• Generar y visualizar reportes de la información disponible en la plataforma</li> <li>• Testing</li> <li>• Integración</li> </ul>	<p>5</p> <p>8</p> <p>8</p> <p>5</p> <p>5</p>

**Tabla 39. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU006:</b> Consultar Reportes
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 23/03/2017	<b>Fecha Fin:</b> 24/03/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Crear interfaces gráficas para visualizar información de los reportes: creación de botones, etiquetas y campos de texto que permitan la consulta de reportes por anteproyecto, proyecto y directores de proyectos

**Tabla 40. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU006:</b> Consultar Reportes
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 8 Horas	
<b>Fecha Inicio:</b> 24/03/2017	<b>Fecha Fin:</b> 27/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso consultar reportes	

**Tabla 41. Task card: Generar y visualizar reportes de la información disponible en la plataforma**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU006:</b> Consultar Reportes

<b>Nombre de Tarea:</b> Generar y visualizar reportes de la información disponible en la plataforma	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 8 Horas	
<b>Fecha Inicio:</b> 27/03/2017	<b>Fecha Fin:</b> 28/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear los métodos que permitan la visualización de la información consultada por el administrador y además crea el reporte consultado en un archivo .pdf	

**Tabla 42. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU006:</b> Consultar Reportes
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 28/03/2017	<b>Fecha Fin:</b> 29/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	



**Tabla 43. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU006:</b> Consultar Reportes
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 29/03/2017	<b>Fecha Fin:</b> 29/03/2017
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).</p>	

**Tabla 44. Historia de Usuario Consultar Información Miembro Comité.**

<b>Historia de usuario</b>
<b>Número:</b> HU007
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Consultar Información Miembro Comité
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 5 puntos de historia
<b>Tiempo estimado:</b> 29 Horas
<b>Perspectiva del producto</b>

**Como:** Usuario Miembro de Comité

**Requiero:** Consultar información correspondiente a: proyectos disponibles y proyectos en el banco de proyectos.

**Para:** poder escoger un proyecto para revisión.

**Criterios de aceptación**

1. Consultar información: requiere que el usuario miembro de comité esté autenticado, y tenga un proyecto en curso. Una vez validada la información el usuario podrá visualizar toda la información de los anteproyectos disponibles y proyectos en el banco de proyectos.
2. Consultar información exitoso: en caso de que toda la información fue validada la plataforma desplegara la información solicitada por el usuario.
3. Consultar información no exitoso: en caso de que no exista la información disponible la plataforma informará al usuario que no se puede mostrar la información.
4. El banco de proyectos lo puede visualizar sin necesidad de estar autenticado en la plataforma

<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"><li>• Crear interfaces</li></ul>	6
<ul style="list-style-type: none"><li>• Conexión al servidor</li></ul>	6
<ul style="list-style-type: none"><li>• Consultar información</li></ul>	7
<ul style="list-style-type: none"><li>• Testing</li></ul>	4
<ul style="list-style-type: none"><li>• Integración</li></ul>	6

**Tabla 45. Task Card: Crear Interfaces**

**TAREA DE INGENIERÍA**

<b>Número de Tarea:</b> 1	<b>HU007:</b> Consultar Información Miembro Comité
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 30/03/2017	<b>Fecha Fin:</b> 30/03/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaces gráficas para visualizar información en la plataforma: creación de botones, etiquetas y campos de texto que permitan la consulta y visualización de la información para un miembro de comité	

**Tabla 46. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU007:</b> Consultar Información Miembro Comité
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 30/03/2017	<b>Fecha Fin:</b> 31/03/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso consultar información de los miembros del comité.

**Tabla 47. Task Card: Consultar información**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU007:</b> Consultar Información Miembro Comité
<b>Nombre de Tarea:</b> Consultar información	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 7 Horas	
<b>Fecha Inicio:</b> 31/03/2017	<b>Fecha Fin:</b> 01/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear los métodos que permitan la visualización de la información consultada por el miembro de comité.	

**Tabla 48. Task card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU007:</b> Consultar Información Miembro Comité
<b>Nombre de Tarea:</b> Testing	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 01/04/2017	<b>Fecha Fin:</b> 01/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 49. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU007:</b> Consultar Información Miembro Comité
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 01/04/2017	<b>Fecha Fin:</b> 02/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

## 4.2.2 Diseño.

A continuación se muestra la interfaz de Registrar Usuario.

Anteproyecto ▾ Proyecto ▾ Cronograma Banco de proyectos Documentos Reportes ▾ Administrar ▾ Cerrar Sesión (ricardo)

← Regresar

Home / Registro

Registro

Nombre \*  
  
Campo requerido

Apellido \*

Identificación

Programa \*  
selecciones una facultad ▾

Nombre de Usuario \*

Email \*

Contraseña \*

Rol

Comite  
 Docente  
 Estudiante  
 Jurado  
 Secretario

Registrar

**Figura 18. Interfaz Registrar Usuario.[Fuente Propia]**

## 4.2.3 Desarrollo.

**Tabla 50. Evento Programado Registro de Usuario**

<b>EVENTO PROGRAMADO REGISTRO DE USUARIO</b>
<b>RESULTADOS DEL REGISTRO</b>

Anteproyecto ▾ Proyecto ▾ Cronograma Banco de proyectos Documentos Reportes ▾ Administrar ▾ Cerrar Sesión (ricardo)

usuario Creado exitosamente

### Registro

Nombre \*

Apellido \*

Identificacion

Programa \*  
selecciones una facultad ▾

Nombre de Usuario \*

Email \*

Contraseña \*

Rol

- Comite
- Docente
- Estudiante
- Jurado
- Secretario

Registrar

## EVENTOS PROGRAMADOS

```

UserController.php
65 public function actionCreate()
66 {
67     $model = new User();
68
69     if ($model->load(Yii::$app->request->post()) && $model->save()) {
70         return $this->redirect(['view', 'id' => $model->id]);
71     } else {
72         return $this->render('create', [
73             'model' => $model,
74         ]);
75     }
76 }
77
78 /**
79  * Updates an existing User model.
80  * If update is successful, the browser will be redirected to the 'view' page
81  * @param integer $id
82  * @return mixed
83  */
84 public function actionUpdate($id)
85 {
86     $model = $this->findModel($id);
87
88     if ($model->load(Yii::$app->request->post()) && $model->save()) {
89         return $this->redirect(['view', 'id' => $model->id]);
90     } else {
91         return $this->render('update', [
92             'model' => $model,

```

#### 4.2.4 Pruebas.

**Tabla 51. Prueba Funcional Registro de Usuario.**

<b>Historia de Usuario</b>	Registrar Usuario.
<b>Propósito</b>	Crear un usuario y asignar un rol.
<b>Requisitos</b>	Ser secretario (administrador).
<b>Pasos</b>	El secretario ingresa al menú en la pestaña de Administrar, ahí debe seleccionar la opción Registrar, en ese momento se desplegará un formulario que pide información personal como nombres, apellidos, identificación, programa, nombre de usuario, email, contraseña, y se elige el rol que tendrá. Finalmente se presiona el botón registrar.
<b>Resultados Esperados</b>	Un mensaje, que da por hecho el registro exitoso.

The screenshot shows a web application interface for user registration. At the top, there is a navigation bar with links: Anteproyecto, Proyecto, Cronograma, Banco de proyectos, Documentos, Reportes, Administrar, and Cerrar Sesión (ricardo). Below the navigation bar is a 'Regresar' button and a breadcrumb 'Home / Registro'. The main content area is titled 'Registro' and contains a form with the following fields:

- Nombre \***: Input field containing 'Andres'.
- Apellido \***: Input field containing 'Cortes'.
- Identificación**: Input field containing '1067287356'.
- Programa \***: Dropdown menu with 'Ing. Sistemas informatico' selected.
- Nombre de Usuario \***: Input field containing 'andres08'.
- Email \***: Input field containing 'andres@uniautonomia.edu.co'.
- Contraseña \***: Input field with masked characters '\*\*\*\*\*'.
- Rol**: Radio button options:
  - Comite
  - Docente
  - Estudiante
  - Jurado
  - Secretario

At the bottom of the form is a blue 'Registrar' button.

**Figura 19. Prueba Funcional Interfaz Registrar Usuario. [Fuente Propia]**



Anteproyecto - Proyecto - Cronograma - Banco de proyectos - Documentos - Reportes - Administrar - Cerrar Sesión (ricardo)

usuario Creado exitosamente

**Registro**

Nombre \*

Apellido \*

Identificacion

Programa \*  
selecciones una facultad

Nombre de Usuario \*

Email \*

Contraseña \*

Rol

- Comite
- Docente
- Estudiante
- Jurado
- Secretario

Registrar

**Figura 20. Prueba Funcional Registrar Usuario. [Fuente Propia]**

#### **4.2.5 Sprint Review.**

Se entregó funcionalidades autenticar usuario, registrar usuario, seleccionar archivo, consultar información estudiante, cambiar contraseña, consultar reportes, consultar información miembro comité, completamente finalizadas, es decir software funcional.

#### **4.2.6 Retrospectiva.**

Se hizo uso de la técnica de la estrella de mar, ya que permite tener retroalimentación, y acorde a esto se ajusta la ejecución de actividades para el siguiente sprint.

**Tabla 52. Retrospectiva del Sprint 1.**

<p><b>Comenzar a Hacer.</b></p> <ul style="list-style-type: none"> <li>• Retroalimentación del proceso de desarrollo.</li> <li>• Usar Atom para la codificación.</li> <li>• Usar el repositorio Bitbucket.</li> </ul>	<p><b>Más de.</b></p> <ul style="list-style-type: none"> <li>• Reuniones del equipo.</li> <li>• Tiempo de aprendizaje.</li> <li>• Estimación del proyecto.</li> </ul>
<p><b>Seguir Haciendo.</b></p> <ul style="list-style-type: none"> <li>• Reuniones con el cliente, para continuo levantamiento y revisión de requisitos.</li> </ul>	<p><b>Menos de.</b></p> <ul style="list-style-type: none"> <li>• Tiempo dedicado al diseño.</li> <li>• Tiempo dedicado a la recodificación.</li> </ul>
<p><b>Dejar de Hacer.</b></p> <ul style="list-style-type: none"> <li>• Pausas prolongadas durante las reuniones del equipo.</li> <li>• Posponer reuniones del equipo.</li> </ul>	

### **4.3 IMPLEMENTACIÓN DEL SPRINT 2.**

Se encuentra conformado por 6 historias de usuario, las cuales se estimaron con un tiempo de ejecución de 16 días (hábiles) de 8 horas de trabajo cada uno, se realizó planeación, diseño, desarrollo y pruebas.

Estas son:

- Consultar Información Jurado.
- Recibir Alerta.
- Consultar Información Docente.
- Rechazar Proyecto
- Asignar Jurados.
- Asignar Fecha de Sustentación.

#### **4.3.1 Planificación.**

A continuación se muestran las historias de usuario y task card del Sprint 2.

**Tabla 53. Historia de usuario Consultar información Jurado**

<b>Historia de usuario</b>	
<b>Número:</b> HU008	
<b>Programador Responsable:</b> Team development	
<b>Nombre de la historia:</b> Consultar Información Jurado	
<b>Nivel de riesgo de desarrollo:</b> Media	
<b>Peso de la historia:</b> 3 puntos de historia	
<b>Tiempo estimado:</b> 14 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario Jurado	
<b>Requiero:</b> Consultar información correspondiente a: proyectos asignados	
<b>Para:</b> poder estudiar y revisar con anterioridad los proyectos.	
<b>Criterios de aceptación</b>	
<ol style="list-style-type: none"> <li>1. Consultar información: requiere que el usuario jurado esté autenticado, que tenga por lo menos un proyecto asignado y que el estado del proyecto sea aprobado. Una vez validada la información el usuario podrá visualizar toda la información de los proyectos disponibles y proyectos en el banco de proyectos.</li> <li>2. Consultar información exitoso: en caso de que toda la información fue validada la plataforma desplegara la información solicitada por el usuario.</li> <li>3. Consultar información no exitoso: en caso de que no exista la información disponible la plataforma informará al usuario que no se puede mostrar la información.</li> </ol>	
<b>Actividades</b>	<b>N° de horas de implementación</b>

• Crear interfaces	3
• Conexión al servidor	3
• Consulta de información	2
• Testing	4
• Integración	2

**Tabla 54. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea: 1</b>	<b>HU008: Consultar Información Jurado</b>
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/04/2017	<b>Fecha Fin:</b> 11/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaces gráficas para visualizar información en la plataforma: creación de botones, etiquetas y campos de texto que permitan la consulta y visualización de la información para un jurado.	

**Tabla 55. Task card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea: 2</b>	<b>HU008: Consultar Información Jurado</b>

<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 11/04/2017	<b>Fecha Fin:</b> 11/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso consultar información de los jurados.	

**Tabla 56. Task Card: Consultar información**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU008:</b> Consultar Información Jurado
<b>Nombre de Tarea:</b> Consultar información	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 11/04/2017	<b>Fecha Fin:</b> 11/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear los métodos que permitan la visualización de la información consultada por el jurado.	

**Tabla 57. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU008:</b> Consultar Información Jurado
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 12/04/2017	<b>Fecha Fin:</b> 12/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 58. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU008:</b> Consultar Información Jurado
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 12/04/2017	<b>Fecha Fin:</b> 12/04/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).

**Tabla 59. Historia de usuario Recibir alerta**

<b>Historia de usuario</b>
<b>Número:</b> HU009
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Recibir alerta
<b>Nivel de riesgo de desarrollo:</b> Alta
<b>Peso de la historia:</b> 8 puntos de historia
<b>Tiempo estimado:</b> 48 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario <b>Requiero:</b> Ser avisado por la plataforma a mi correo electrónico con anterioridad acerca del tiempo para realizar tareas correspondientes a cada rol. <b>Para:</b> Poder cumplir con los tiempos establecidos en la resolución 0047.
<b>Criterios de aceptación</b>
1. Para recibir una alerta el usuario debe estar registrado en la plataforma, se visualizará una notificación en su perfil y también recibirá un correo electrónico avisándole de la alerta que recibió.

<p>2. Las alertas se programan y envían de acuerdo al cronograma de tiempos establecidos según la resolución 0047.</p> <p>3. La notificación es visible hasta que el usuario la revise, después de esto desaparece de la plataforma.</p>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor</li> <li>• Crear sistema de alertas</li> <li>• Testing</li> <li>• Integración</li> </ul>	<p>2</p> <p>10</p> <p>20</p> <p>8</p> <p>8</p>

**Tabla 60. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU009:</b> Recibir Alerta
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 12/04/2017	<b>Fecha Fin:</b> 12/04/2017
<b>Programador Responsable:</b> Team Development	



**Descripción:**

Crear interfaces gráficas para enviar y recibir notificaciones

**Tabla 61. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU009:</b> Recibir Alerta
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 10 Horas	
<b>Fecha Inicio:</b> 13/04/2017	<b>Fecha Fin:</b> 14/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso para recibir alerta.	

**Tabla 62. Task Card: Crear Sistema de Alertas**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU009:</b> Recibir Alerta
<b>Nombre de Tarea:</b> Crear Sistema de Alertas	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia

<b>Tiempo de implementación estimado:</b> 20 Horas	
<b>Fecha Inicio:</b> 14/04/2017	<b>Fecha Fin:</b> 18/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear los métodos que permitan realizar el envío y recepción de las notificaciones para los diferentes perfiles en la plataforma. Además de la funcionalidad que permita enviar mensajes por Email.	

**Tabla 63. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU009:</b> Recibir Alerta
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 8 Horas	
<b>Fecha Inicio:</b> 18/04/2017	<b>Fecha Fin:</b> 19/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 64. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU009:</b> Recibir Alerta
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 8 puntos de historia
<b>Tiempo de implementación estimado:</b> 8 Horas	
<b>Fecha Inicio:</b> 19/04/2017	<b>Fecha Fin:</b> 20/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 65. Historia de usuario Consultar información Docente**

<b>Historia de usuario</b>
<b>Número:</b> HU010
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Consultar Información Docente
<b>Nivel de riesgo de desarrollo:</b> Media
<b>Peso de la historia:</b> 3 puntos de historia
<b>Tiempo estimado:</b> 13 Horas

### **Perspectiva del producto**

**Como:** Usuario Docente

**Requiero:** Consultar información correspondiente a cronograma de actividades, banco de proyectos y documentos de interés.

**Para:** poder conocer actividades y/o propuestas de proyectos a realizar.

### **Criterios de aceptación**

1. Consultar información: requiere que el usuario docente esté autenticado. Una vez validada la información el usuario podrá consultar y visualizar la información del cronograma de actividades, del banco de proyectos y documentos de interés.
2. Consultar información exitoso: en caso de que toda la información fue validada la plataforma desplegará la información solicitada por el usuario.
3. Consultar información no exitoso: en caso de que no exista la información disponible la plataforma informará al usuario que no se puede mostrar la información.
4. El usuario docente podrá acceder al banco de proyectos sin autenticarse.

<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"><li>• Crear interfaces</li></ul>	3
<ul style="list-style-type: none"><li>• Conexión al servidor</li></ul>	3
<ul style="list-style-type: none"><li>• Consulta información</li></ul>	2
<ul style="list-style-type: none"><li>• Testing</li></ul>	3
<ul style="list-style-type: none"><li>• Integración</li></ul>	2

**Tabla 66. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU010:</b> Consultar Información Docente
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 20/04/2017	<b>Fecha Fin:</b> 21/04/2017
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>            Crear interfaces gráficas para visualizar información en la plataforma: creación de botones, etiquetas y campos de texto que permitan la consulta y visualización de la información para un docente.</p>	

**Tabla 67. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU010:</b> Consultar Información Docente
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 21/04/2017	<b>Fecha Fin:</b> 21/04/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso consultar información de los docentes.

**Tabla 68. Task Card: Consultar información**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU010:</b> Consultar Información Docente
<b>Nombre de Tarea:</b> Consultar información	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 21/04/2017	<b>Fecha Fin:</b> 21/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear los métodos que permitan la visualización de la información consultada por el docente.	

**Tabla 69. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU010:</b> Consultar Información Docente

<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 21/04/2017	<b>Fecha Fin:</b> 24/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 70. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU010:</b> Consultar Información Docente
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 24/04/2017	<b>Fecha Fin:</b> 24/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 71. Historia de usuario Rechazar proyecto**

<b>Historia de usuario</b>
<b>Número:</b> HU011
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Rechazar proyecto
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 5 puntos de historia
<b>Tiempo estimado:</b> 17 Horas
<p style="text-align: center;"><b>Perspectiva del producto</b></p> <p><b>Como:</b> Usuario Administrador</p> <p><b>Requiero:</b> Rechazar un proyecto</p> <p><b>Para:</b> Modificar el estado del proyecto a: Rechazado, e informar al estudiante que el proyecto no cumplió con los criterios de calificación exigidos.</p>
<p style="text-align: center;"><b>Criterios de aceptación</b></p> <ol style="list-style-type: none"><li>1. Rechazar proyecto requiere que el usuario administrador esté autenticado en la plataforma, después de recibir un concepto del comité se actualiza el estado del proyecto y se fija en rechazado. Una vez realizado este proceso se notifica al estudiante del estado de su proyecto y el concepto emitido.</li><li>2. Rechazar proyecto exitoso: la plataforma muestra un mensaje de éxito informando que la acción se realizó satisfactoriamente.</li><li>3. Rechazar proyecto no exitoso: la plataforma muestra un mensaje de error informando que la acción no se pudo realizar y permite volver a</li></ol>



intentarlo.	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor</li> <li>• Cambiar estado de proyecto</li> <li>• Testing</li> <li>• Integración</li> </ul>	3 6 3 3 2

**Tabla 72. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU011:</b> Rechazar Proyecto
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 24/04/2017	<b>Fecha Fin:</b> 24/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para que el usuario Administrador modifique el estado de un proyecto: etiquetas, campos de texto y botones que permitan realizar la modificación del estado de un proyecto a: rechazado.	

**Tabla 73. Task Card: Conexión al Servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU011:</b> Rechazar Proyecto
<b>Nombre de Tarea:</b> Conexión al Servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 24/04/2017	<b>Fecha Fin:</b> 25/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear una conexión entre el servidor y la aplicación web para que responda a la solicitudes realizadas por el usuario para poder autenticarse en la plataforma.	

**Tabla 74. Task Card: Cambiar Estado de un proyecto**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU011:</b> Rechazar Proyecto
<b>Nombre de Tarea:</b> Cambiar Estado de un proyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 25/04/2017	<b>Fecha Fin:</b> 26/04/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Crear un método para consultar y actualizar la base de datos, para así permitir que el usuario Administrador modifique el estado de un proyecto y fijarlo en rechazado.

**Tabla 75. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU011:</b> Rechazar Proyecto
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 26/04/2017	<b>Fecha Fin:</b> 26/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 76. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU011:</b> Rechazar Proyecto
<b>Nombre de Tarea:</b> Integración	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 26/04/2017	<b>Fecha Fin:</b> 26/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 77. Historia de usuario Asignar Jurados**

<b>Historia de usuario</b>
<b>Número:</b> HU012
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Asignar Jurados
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 3 puntos de historia
<b>Tiempo estimado:</b> 16 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario Administrador
<b>Requiero:</b> Asignar jurados
<b>Para:</b> evaluación del proyecto de grado

### Criterios de aceptación

1. Asignar Jurados requiere que el usuario administrador esté autenticado en la plataforma, una vez se verifica que el estado del proyecto es aceptado y que ya esté creada la lista de jurados, el administrador puede escoger los docentes que desea asignar al proyecto como evaluadores.
2. Asignar Jurados exitoso: la plataforma muestra un mensaje de éxito informando que la acción se realizó satisfactoriamente y se envía una alerta a los estudiantes a quienes pertenece el proyecto.
3. Asignar Jurados no exitoso: la plataforma muestra un mensaje de error informando que no se pudo realizar la acción y permite volverlo a intentar.

Actividades	N° de horas de implementación
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor</li> <li>• Crear asignación de jurados.</li> <li>• Testing</li> <li>• Integración</li> </ul>	<p>3</p> <p>6</p> <p>3</p> <p>2</p> <p>2</p>

**Tabla 78. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU012:</b> Asignar Jurado
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia

<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 26/04/2017	<b>Fecha Fin:</b> 27/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para asignar jurado: crear etiquetas, despleables y botones que permitan realizar al usuario la asignación de jurados (jurado, jurado2, proyecto).	

**Tabla 79. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU012:</b> Asignar Jurado
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 27/04/2017	<b>Fecha Fin:</b> 27/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso asignar jurado.	

**Tabla 80. Task Card: Crear asignar jurado**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU012:</b> Asignar Jurado
<ul style="list-style-type: none"> <li><b>Nombre de Tarea:</b> Crear asignar Jurado</li> </ul>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 27/04/2017	<b>Fecha Fin:</b> 28/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita la selección de los jurados para ser asignados a un proyecto	

**Tabla 81. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU012:</b> Asignar Jurado
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 28/04/2017	<b>Fecha Fin:</b> 28/04/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.

**Tabla 82. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU012:</b> Asignar Jurado
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 28/04/2017	<b>Fecha Fin:</b> 28/04/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 83. Historia de usuario Asignar Fecha de Sustentación**

<b>Historia de usuario</b>
<b>Número:</b> HU013
<b>Programador Responsable:</b> Team development



<b>Nombre de la historia:</b> Asignar Fecha de Sustentación	
<b>Nivel de riesgo de desarrollo:</b> media	
<b>Peso de la historia:</b> 3 puntos de historia	
<b>Tiempo estimado:</b> 20 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario Administrador	
<b>Requiero:</b> Asignar fecha de Sustentación	
<b>Para:</b> Informar al Usuario estudiante el día y la hora para la sustentación del proyecto de grado.	
<b>Criterios de aceptación</b>	
<ol style="list-style-type: none"> <li>1. Asignar fecha de sustentación requiere que el usuario Administrador este autenticado, una vez se ha verificado que el estado del proyecto sea: Aceptado, se consulta el cronograma para realizar la asignación de fecha, hora y lugar para la sustentación.</li> <li>2. Asignar fecha de sustentación exitoso: la plataforma muestra un mensaje de éxito informando que la acción se realizó satisfactoriamente y se notifica a los estudiantes y a los jurados de la fecha correspondiente.</li> <li>3. Asignar fecha de sustentación no exitoso: la plataforma muestra un mensaje de error informando que la acción no se pudo realizar y permite volver a intentar.</li> </ol>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor</li> <li>• Consultar cronograma</li> </ul>	<p>3</p> <p>5</p> <p>4</p>

<ul style="list-style-type: none"> <li>• Crear fecha de sustentación</li> </ul>	3
<ul style="list-style-type: none"> <li>• Testing</li> </ul>	3
<ul style="list-style-type: none"> <li>• Integración</li> </ul>	2

**Tabla 84. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU013:</b> Asignar fecha de sustentación
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 28/04/2017	<b>Fecha Fin:</b> 01/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para que el usuario Administrador consulte el cronograma de actividades y realice la asignación de una fecha de sustentación para un proyecto.	

**Tabla 85. Task Card: Conexión al Servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU013:</b> Asignar fecha de sustentación

<b>Nombre de Tarea:</b> Conexión al Servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 01/05/2017	<b>Fecha Fin:</b> 01/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear una conexión entre el servidor y la aplicación web para que responda a la solicitudes realizadas por el usuario para poder asignar fecha de sustentación.	

**Tabla 86. Task Card: Consultar Cronograma**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU013:</b> Asignar fecha de sustentación
<b>Nombre de Tarea:</b> Consultar Cronograma	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 01/05/2017	<b>Fecha Fin:</b> 02/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita al usuario Administrador consultar el cronograma de actividades para saber en qué fecha puede ser asignada la sustentación de un proyecto.	

**Tabla 87. Task Card: Crear fecha de sustentación**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU013:</b> Asignar fecha de sustentación
<b>Nombre de Tarea:</b> Crear fecha de sustentación	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 02/05/2017	<b>Fecha Fin:</b> 02/05/2017
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>            Crear un método que permita al usuario Administrador asignar una fecha de sustentación de un proyecto después de haber consultado el cronograma de actividades, con fecha, hora y lugar.</p>	

**Tabla 88. Task Card: Testing**

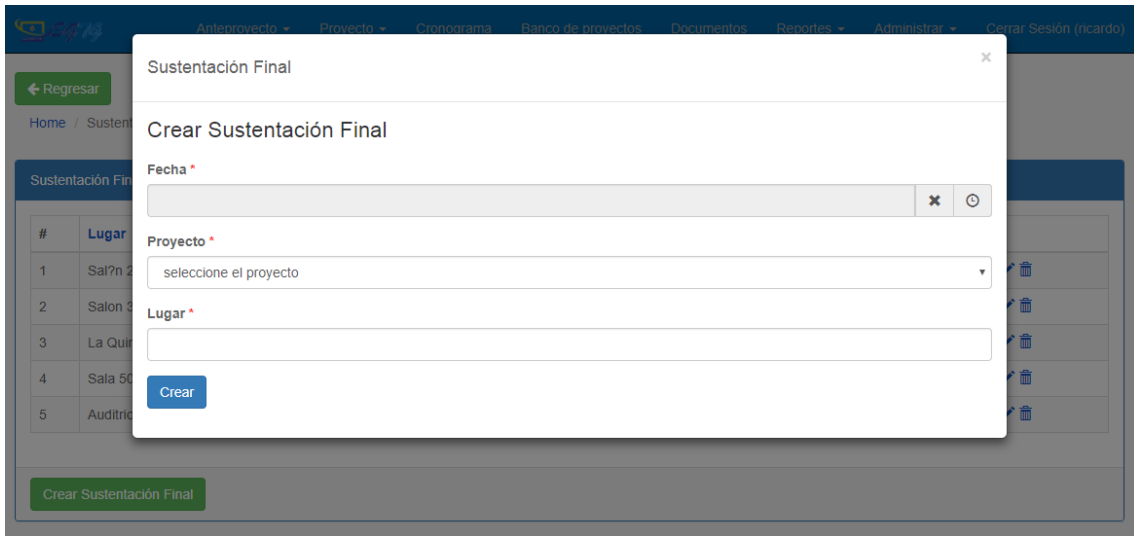
<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU013:</b> Asignar fecha de sustentación
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 02/05/2017	<b>Fecha Fin:</b> 02/05/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.

**Tabla 89. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 6	<b>HU013:</b> Asignar fecha de sustentación
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 03/05/2017	<b>Fecha Fin:</b> 03/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

### 4.3.2 Diseño.



**Figura 21. Interfaz Asignar Fecha de Sustentación Final. [Fuente Propia]**

### 4.3.3 Desarrollo.

**Tabla 90. Evento Asignar Fecha de Sustentación**

EVENTO PROGRAMADO ASIGNAR FECHA DE SUSTENTACIÓN FINAL	
RESULTADOS DE LA ASIGNACIÓN	
EVENTOS PROGRAMADOS	

```

    @return mixed
    */
    public function actionCreate()
    {
        $model = new SustentacionFinal();

        if ($model->load(Yii::$app->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->idsustentacion_final]);
        } else {
            return $this->renderAjax('create', [
                'model' => $model,
            ]);
        }
    }

    /**
     * Updates an existing SustentacionFinal model.
     * If update is successful, the browser will be redirected to the 'view' page.
     * @param integer $id
     * @return mixed
     */
    public function actionUpdate($id)
    {
        $model = $this->findModel($id);

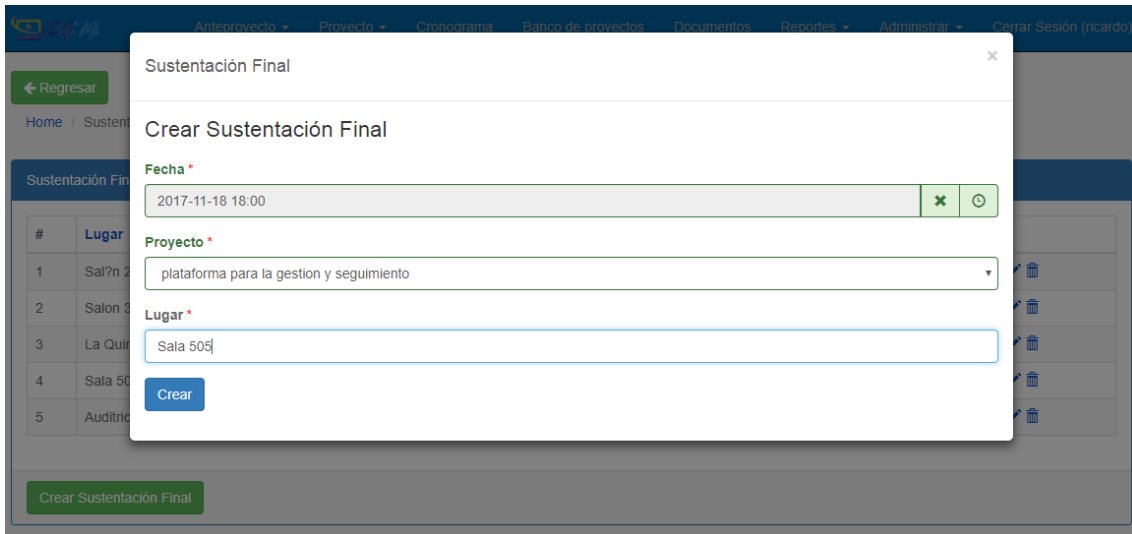
        if ($model->load(Yii::$app->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->idsustentacion_final]);
        } else {
            return $this->render('update', [

```

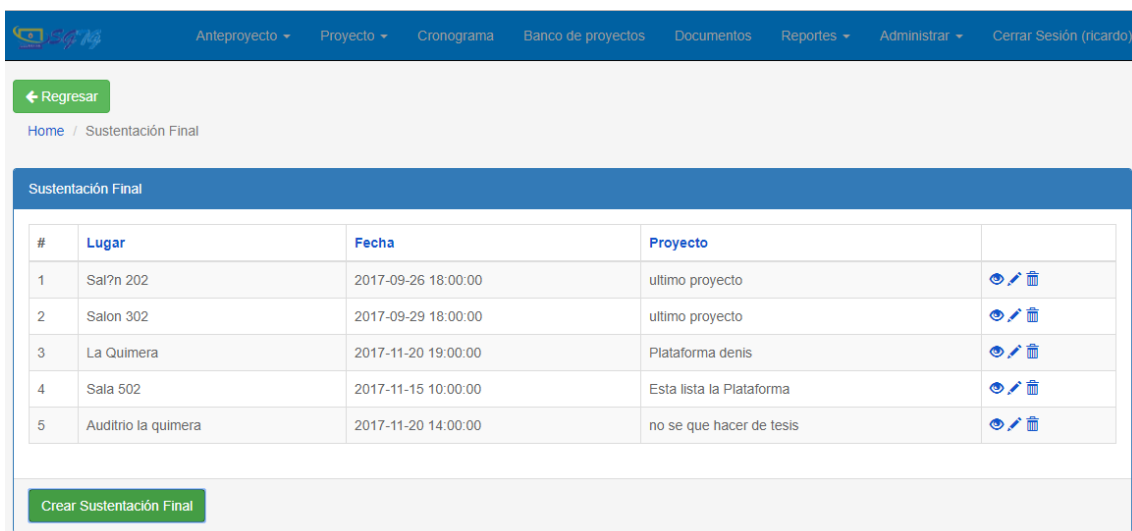
#### 4.3.4 Pruebas.

**Tabla 91. Prueba Funcional Asignar Sustentación Final.**

<b>Historia de Usuario</b>	Asignar Fecha de Sustentación.
<b>Propósito</b>	Asignar fecha de sustentación a los proyectos aprobados.
<b>Requisitos</b>	Ser secretario (administrador).
<b>Pasos</b>	El secretario ingresa al menú en la pestaña de Proyecto, ahí debe seleccionar la opción Asignar Sustentación, en ese momento aparecerá un listado que contiene todas las fechas de sustentaciones, luego se desplegará un formulario con estos campos; fecha, proyecto y lugar. Finalmente se presiona el botón crear.
<b>Resultados Esperados</b>	Asigne la fecha de la sustentación y la envíe al estudiante.

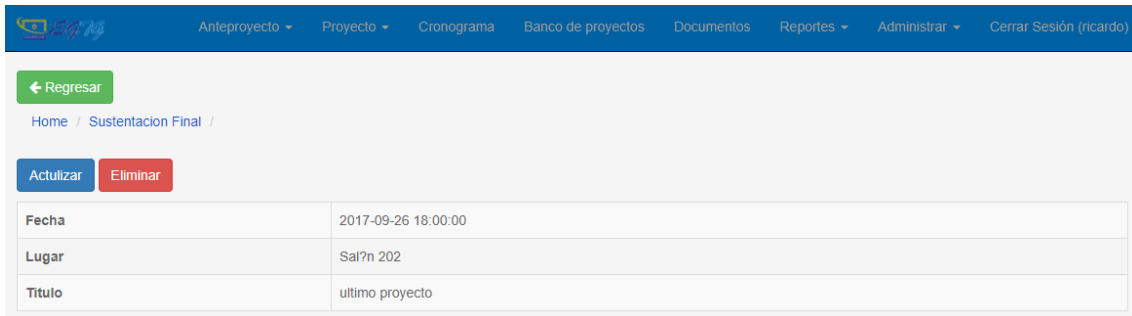


**Figura 22. Interfaz Asignar Fechas de sustentación Final. [Fuente Propia]**

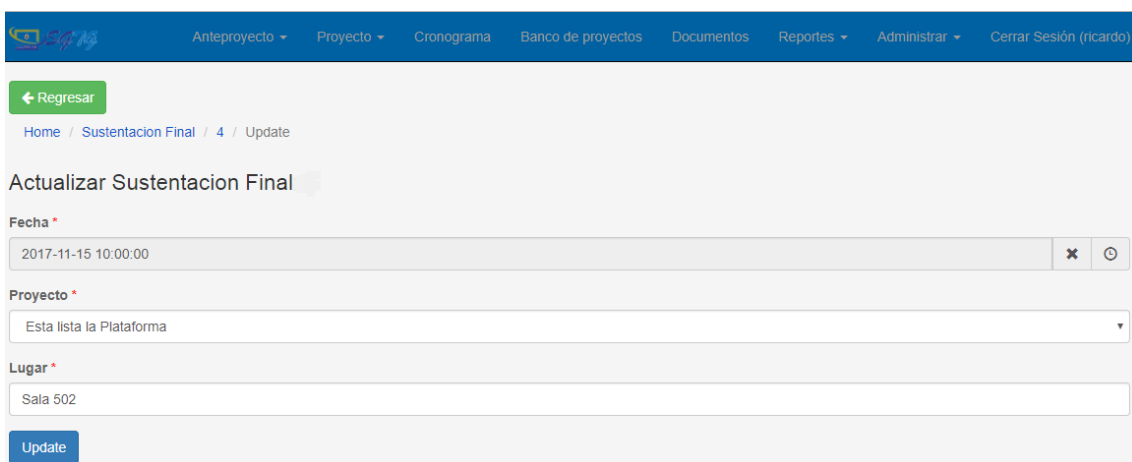


**Figura 23. Interfaz Prueba de Asignar Fecha de Sustentación. [Fuente Propia]**





**Figura 24. Interfaz Ver detalles. [Fuente Propia]**



**Figura 25. Interfaz Actualizar Sustentación Final. [Fuente Propia]**

#### 4.3.5 Sprint Review.

Se entregó funcionalidades consultar información jurado, recibir alerta, consultar información docente, rechazar proyecto, asignar jurados, asignar fecha de sustentación, completamente finalizadas, es decir, software funcional.

#### 4.3.6 Retrospectiva.

Se hizo uso de la técnica de la estrella de mar, ya que permite tener retroalimentación, y acorde a esto se ajusta la ejecución de actividades para el siguiente sprint.

**Tabla 92. Retrospectiva del Sprint 2.**

<p><b>Comenzar a Hacer.</b></p> <ul style="list-style-type: none"> <li>• Utilizar los medios tecnológicos para la comunicación del equipo.</li> </ul>	<p><b>Más de.</b></p> <ul style="list-style-type: none"> <li>• Compartir la documentación encontrada con el equipo de desarrollo.</li> </ul>
<p><b>Seguir Haciendo.</b></p> <ul style="list-style-type: none"> <li>• Reuniones con el cliente.</li> <li>• Reuniones del equipo.</li> <li>• Retroalimentación del proceso de desarrollo.</li> </ul>	<p><b>Menos de.</b></p> <ul style="list-style-type: none"> <li>• Documentación excesiva.</li> </ul>
<p><b>Dejar de Hacer.</b></p> <ul style="list-style-type: none"> <li>• Pequeñas charlas que distraigan al equipo.</li> <li>• Documentación Innecesaria.</li> </ul>	

#### **4.4 IMPLEMENTACIÓN DEL SPRINT 3.**

Se encuentra conformado por 7 historias de usuario, las cuales se estimaron con un tiempo de ejecución de 16 días (hábiles) de 8 horas de trabajo cada uno, se realizó planeación, diseño, desarrollo y pruebas.

Estas son:

- Crear Jurado.
- Recuperar Contraseña.
- Crear Anteproyecto.
- Crear Proyecto.
- Asignar Concepto.
- Crear Director Proyecto.
- Crear Revisión.

#### 4.4.1 Planificación.

A continuación se muestran las historias de usuario y task card del Sprint 3.

**Tabla 93. Historia de usuario Crear Jurado**

Historia de usuario
<b>Número:</b> HU014
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Crear Jurado
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 3 puntos de historia
<b>Tiempo estimado:</b> 18 Horas
<p style="text-align: center;"><b>Perspectiva del producto</b></p> <p><b>Como:</b> Usuario Administrador</p> <p><b>Requiero:</b> Crear jurado</p> <p><b>Para:</b> Poder crear una lista de posibles jurados para ser asignados posteriormente.</p>
<p style="text-align: center;"><b>Criterios de aceptación</b></p> <ol style="list-style-type: none"><li>1. Crear jurado requiere que el usuario administrador este autenticado en la plataforma, el administrador ingresa los datos (nombre, correo) de los docentes que pueden ser jurados de evaluación de proyectos y los almacena en la plataforma para que después puedan ser asignados.</li><li>2. Crear jurado exitoso: la plataforma muestra un mensaje de éxito informando que la acción se registró satisfactoriamente y muestra la lista con el nuevo registro realizado.</li><li>3. Crear jurado no exitoso: la plataforma muestra un mensaje de error</li></ol>

informando que la acción no se pudo realizar y permite volver a intentarlo.	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor</li> <li>• Crear jurado</li> <li>• Testing</li> <li>• Integración</li> </ul>	<p>3</p> <p>5</p> <p>4</p> <p>3</p> <p>3</p>

**Tabla 94. Task Card Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU014:</b> Crear Jurado
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 15/05/2017	<b>Fecha Fin:</b> 15/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para crear jurado: crear etiquetas, campos de texto y botones que permitan realizar al usuario la creación de jurados (nombre de jurado).	

**Tabla 95. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU014:</b> Crear Jurado
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 15/05/2017	<b>Fecha Fin:</b> 15/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso crear jurado.	

**Tabla 96. Task Card: Crear jurado**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU014:</b> Crear Jurado
<ul style="list-style-type: none"> <li>• <b>Nombre de Tarea:</b> Crear Jurado</li> </ul>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 16/05/2017	<b>Fecha Fin:</b> 16/05/2017

<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Crear un método que permita la creación de los jurados en una lista para ser asignados posteriormente

**Tabla 97. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU014:</b> Crear Jurado
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 16/05/2017	<b>Fecha Fin:</b> 16/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 98. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU014:</b> Crear Jurado
<b>Nombre de Tarea:</b> Integración	

<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 16/05/2017	<b>Fecha Fin:</b> 17/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 99. Historia de usuario Recuperar Contraseña**

<b>Historia de usuario</b>
<b>Número:</b> HU015
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Recuperar Contraseña
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 2 puntos de historia
<b>Tiempo estimado:</b> 19 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario
<b>Requiero:</b> Recuperar la contraseña
<b>Para:</b> Poder ingresar a la plataforma en caso de haber olvidado la contraseña

**Criterios de aceptación**

1. Recuperar Contraseña: requiere que el usuario esté registrado en la plataforma. Cuando el usuario requiere recuperar su contraseña ingresa su dirección de correo electrónico a donde el sistema enviará un link en donde se permite realizar la actualización de la contraseña.
2. Recuperar contraseña exitoso: Si el usuario está registrado en la plataforma y su correo es correcto, se envía un mensaje informando que se ha enviado los datos de recuperación al correo electrónico.
3. Recuperar contraseña no exitoso: Se envía un mensaje de error informando al usuario que no se puede realizar la operación porque el correo no se encuentra registrado, o el correo no es válido.

<b>Actividades</b>	<b>N° de horas de implementación</b>
• Crear interfaces	3
• Conexión al servidor	4
• Crear recuperación de contraseña	6
• Testing	2
• Integración	4

**Tabla 100. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU015:</b> Recuperar contraseña
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	



<b>Fecha Inicio:</b> 17/05/2017	<b>Fecha Fin:</b> 17/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para recuperar contraseña: crear etiquetas, campos de texto y botones (Email).	

**Tabla 101. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU015:</b> Recuperar contraseña
<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 17/05/2017	<b>Fecha Fin:</b> 18/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso recuperar contraseña.	

**Tabla 102. Task Card: Crear recuperación de contraseña**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU015:</b> Recuperar contraseña

<b>Nombre de Tarea:</b> Crear recuperación de contraseña	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 18/05/2017	<b>Fecha Fin:</b> 18/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita consultar si el usuario está registrado en la plataforma y enviar como respuesta un Email con un link que habilita el cambio de contraseña.	

**Tabla 103. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU015:</b> Recuperar contraseña
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 18/05/2017	<b>Fecha Fin:</b> 19/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 104. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU015:</b> Recuperar contraseña
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 2 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 19/05/2017	<b>Fecha Fin:</b> 19/05/2017
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b>                      Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).</p>	

**Tabla 105. Historia de usuario Crear Anteproyecto**

<b>Historia de usuario</b>
<b>Número:</b> HU016
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Crear Anteproyecto
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 5 puntos de historia
<b>Tiempo estimado:</b> 19 Horas

### Perspectiva del producto

**Como:** Usuario Estudiante

**Requiero:** Crear el anteproyecto

**Para:** Poder enviar el anteproyecto y empezar con la gestión y seguimiento del mismo.

### Criterios de aceptación

1. Crear Anteproyecto requiere que el usuario estudiante esté autenticado en la plataforma, el estudiante ingresa los datos (título del anteproyecto, descripción, archivo de anteproyecto, fecha de creación, modalidad) necesarios para crear un anteproyecto.
2. Crear Anteproyecto exitoso: la plataforma muestra un mensaje de éxito informando que la acción se registró satisfactoriamente y se puede visualizar el anteproyecto registrado.
3. Crear Anteproyecto no exitoso: si el usuario no ingresa correctamente los datos requeridos, la plataforma muestra un mensaje de error informando que la acción no se pudo realizar y permite volver a intentarlo.

Actividades	N° de horas de implementación
<ul style="list-style-type: none"><li>• Crear interfaces</li></ul>	4
<ul style="list-style-type: none"><li>• Conexión al servidor</li></ul>	5
<ul style="list-style-type: none"><li>• Crear anteproyecto</li></ul>	5
<ul style="list-style-type: none"><li>• Testing</li></ul>	2
<ul style="list-style-type: none"><li>• Integración</li></ul>	3

**Tabla 106. Task Card: Crear Interfaces**

**TAREA DE INGENIERÍA**

<b>Número de Tarea:</b> 1	<b>HU016:</b> Crear Anteproyecto
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 19/05/2017	<b>Fecha Fin:</b> 22/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para que el usuario Estudiante cree un anteproyecto: etiquetas, campos de texto y botones que permitan realizar la creación de un anteproyecto (título del anteproyecto, descripción, archivo de anteproyecto, fecha de creación, modalidad).	

**Tabla 107. Task Card: Conexión al Servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU016:</b> Crear Anteproyecto
<b>Nombre de Tarea:</b> Conexión al Servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	

<b>Fecha Inicio:</b> 22/05/2017	<b>Fecha Fin:</b> 22/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear una conexión entre el servidor y la aplicación web para que responda a la solicitudes realizadas por el usuario para poder crear anteproyecto.	

**Tabla 108. Task Card: Crear Anteproyecto**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU016:</b> Crear Anteproyecto
<b>Nombre de Tarea:</b> Crear Anteproyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 22/05/2017	<b>Fecha Fin:</b> 23/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita crear un nuevo registro en la base de datos de un anteproyecto por un usuario Estudiante, con los siguientes campos título del anteproyecto, descripción, archivo de anteproyecto, fecha de creación, modalidad.	

**Tabla 109. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>
----------------------------

<b>Número de Tarea:</b> 3	<b>HU016:</b> Crear Anteproyecto
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 23/05/2017	<b>Fecha Fin:</b> 23/05/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 110. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU016:</b> Crear Anteproyecto
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 23/05/2017	<b>Fecha Fin:</b> 23/05/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).

**Tabla 111. Historia de usuario Crear Proyecto**

<b>Historia de usuario</b>
<b>Número:</b> HU017
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Crear Proyecto
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 5 puntos de historia
<b>Tiempo estimado:</b> 19 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario Estudiante
<b>Requiero:</b> Crear un proyecto
<b>Para:</b> Poder enviar el proyecto y empezar con la gestión y seguimiento del mismo. .
<b>Criterios de aceptación</b>
<ol style="list-style-type: none"> <li>1. Para que el usuario Estudiante tenga habilitada la opción de crear proyecto debe tener el anteproyecto en estado: <b>Aceptado</b></li> <li>2. Crear proyecto requiere que el usuario estudiante esté autenticado en la plataforma, el estudiante ingresa los datos (título del proyecto, descripción, archivo del proyecto, fecha de creación, modalidad)</li> </ol>



necesarios para crear un proyecto.

3. Crear proyecto exitoso: la plataforma muestra un mensaje de éxito informando que la acción se registró satisfactoriamente y se envía una notificación al administrador para informarle que se ha creado un proyecto.
4. Crear proyecto no exitoso: si el usuario no ingresa correctamente los datos requeridos, la plataforma muestra un mensaje de error informando que la acción no se pudo realizar y permite volver a intentarlo..

Actividades	N° de horas de implementación
<ul style="list-style-type: none"> <li>• Crear interfaces</li> </ul>	4
<ul style="list-style-type: none"> <li>• Conexión al servidor</li> </ul>	5
<ul style="list-style-type: none"> <li>• Crear proyecto</li> </ul>	5
<ul style="list-style-type: none"> <li>• Testing</li> </ul>	2
<ul style="list-style-type: none"> <li>• Integración</li> </ul>	3

**Tabla 112. Task Card: Crear Interfaces**

TAREA DE INGENIERÍA	
<b>Número de Tarea:</b> 1	<b>HU017:</b> Crear Proyecto
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 4 Horas	
<b>Fecha Inicio:</b> 02/08/2017	<b>Fecha Fin:</b> 02/08/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Crear interfaz gráfica para que el usuario Estudiante cree un proyecto: etiquetas, campos de texto y botones que permitan realizar la creación de un proyecto (título del proyecto, descripción, archivo del proyecto, fecha de creación, modalidad).

**Tabla 113. Task Card: Conexión al Servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU017:</b> Crear Proyecto
<b>Nombre de Tarea:</b> Conexión al Servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 02/08/2017	<b>Fecha Fin:</b> 02/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear una conexión entre el servidor y la aplicación web para que responda a la solicitudes realizadas por el usuario para poder crear proyecto.	

**Tabla 114. Task Card: Crear Proyecto**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU017:</b> Crear Proyecto

<b>Nombre de Tarea:</b> Crear Proyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 03/08/2017	<b>Fecha Fin:</b> 03/08/2017
<b>Programador Responsable:</b> Team Development	
<p><b>Descripción:</b></p> <p>Crear un método que permita crear un nuevo registro en la base de datos de un proyecto por un usuario Estudiante, con los siguientes campos título del proyecto, descripción, archivo del proyecto, fecha de creación, modalidad.</p> <p>Además, se crea una funcionalidad que permita notificar al administrador por medio de un email que se ha registrado un nuevo proyecto.</p>	

**Tabla 115. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU017:</b> Crear Proyecto
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 03/08/2017	<b>Fecha Fin:</b> 03/08/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.

**Tabla 116. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU017:</b> Crear Proyecto
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 04/08/2017	<b>Fecha Fin:</b> 04/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 117. Historia de usuario Asignar Concepto**

<b>Historia de usuario</b>
<b>Número:</b> HU018
<b>Programador Responsable:</b> Team development

<b>Nombre de la historia:</b> Asignar Concepto	
<b>Nivel de riesgo de desarrollo:</b> media	
<b>Peso de la historia:</b> 5 puntos de historia	
<b>Tiempo estimado:</b> 17 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario Docente Comité	
<b>Requiero:</b> Asignar un concepto para anteproyecto.	
<b>Para:</b> Poder informar al Usuario Estudiante de que su anteproyecto ya ha sido revisado por los miembros del comité y está en un estado de: Aceptado, Rechazado o Corrección.	
<b>Criterios de aceptación</b>	
<ol style="list-style-type: none"> <li>1. El usuario Docente Comité debe estar autenticado cuando requiera asignar un concepto para un anteproyecto que previamente debe estar creado.</li> <li>2. Asignar concepto exitoso: Si el usuario Docente Comité está registrado en la plataforma tendrá la opción de asignar concepto habilitada. Para realizar la asignación debe ingresar los siguientes datos: descripción, corrección, selecciona un archivo que desea cargar con los conceptos emitidos y escoge un anteproyecto para asociarlo al concepto y además elige el estado del anteproyecto. Una vez ingresada toda la información se informa al usuario que la acción se realizó satisfactoriamente.</li> <li>3. Asignar concepto no exitoso: Se envía un mensaje de error informando al usuario que no se puede realizar la operación porque el usuario no ingresó correctamente la información requerida.</li> </ol>	
<b>Actividades</b>	<b>N° de horas de implementación</b>

• Crear interfaces	3
• Conexión al servidor	6
• Crear concepto	3
• Testing	3
• Integración	2

**Tabla 118. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea: 1</b>	<b>HU018: Asignar Concepto</b>
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 04/08/2017	<b>Fecha Fin:</b> 04/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para asignar concepto: crear etiquetas, campos de texto y botones (Descripción, Corrección, Archivo, Estado, Proyecto).	

**Tabla 119. Task Card: conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea: 2</b>	<b>HU018: Asignar Concepto</b>

<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 04/08/2017	<b>Fecha Fin:</b> 07/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso asignar concepto.	

**Tabla 120. Task Card: Crear concepto**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU018:</b> Asignar Concepto
<b>Nombre de Tarea:</b> Crear Concepto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 07/08/2017	<b>Fecha Fin:</b> 07/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita ingresar toda la información que se ha emitido para la creación del concepto.	

**Tabla 121. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU018:</b> Asignar Concepto
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 07/08/2017	<b>Fecha Fin:</b> 08/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 122. Task Card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU018:</b> Asignar Concepto
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 08/08/2017	<b>Fecha Fin:</b> 08/08/2017



<b>Programador Responsable:</b> Team Development
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).

**Tabla 123. Historia de usuario Crear Director Proyecto**

<b>Historia de usuario</b>
<b>Número:</b> HU019
<b>Programador Responsable:</b> Team development
<b>Nombre de la historia:</b> Crear Director Proyecto
<b>Nivel de riesgo de desarrollo:</b> media
<b>Peso de la historia:</b> 3 puntos de historia
<b>Tiempo estimado:</b> 17 Horas
<b>Perspectiva del producto</b>
<b>Como:</b> Usuario Administrador <b>Requiero:</b> Crear un director de proyecto <b>Para:</b> Poder asignar la persona que dirigirá dicho proyecto. .
<b>Criterios de aceptación</b>
1. Crear director de proyecto: requiere que el administrador esté autenticado en la plataforma, el usuario administrador deberá ingresar los datos

<p>necesarios para crear el registro (nombre del director de proyecto) y asignarlo a un proyecto.</p> <p>2. Crear director de proyecto exitoso: si los datos ingresados están correctamente validados, la plataforma envía un mensaje informando que el registro se ha creado satisfactoriamente.</p> <p>3. Crear director de proyecto no exitoso: si los datos ingresados no están validados, la plataforma envía un mensaje informando que no se puede realizar la operación.</p>	
<b>Actividades</b>	<b>N° de horas de implementación</b>
<ul style="list-style-type: none"> <li>• Crear interfaces</li> <li>• Conexión al servidor</li> <li>• Crear director de proyecto</li> <li>• Testing</li> <li>• Integración</li> </ul>	<p>3</p> <p>6</p> <p>3</p> <p>3</p> <p>2</p>

**Tabla 124. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 1	<b>HU019:</b> Crear Director de Proyecto
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	

<b>Fecha Inicio:</b> 08/08/2017	<b>Fecha Fin:</b> 08/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para que el usuario Administrador pueda crear un director de proyecto y asignarlo a un proyecto: etiquetas, campos de texto y botones (nombre del director y proyecto asignado)	

**Tabla 125. Task Card: Conexión al Servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 2	<b>HU019:</b> Crear Director de Proyecto
<b>Nombre de Tarea:</b> Conexión al Servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 08/08/2017	<b>Fecha Fin:</b> 09/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear una conexión entre el servidor y la aplicación web para que responda a la solicitudes realizadas por el usuario para poder crear un director de proyecto y asignarle el proyecto que va a dirigir.	

**Tabla 126. Task Card: Crear Director de Proyecto**

<b>TAREA DE INGENIERÍA</b>
----------------------------

<b>Número de Tarea:</b> 3	<b>HU019:</b> Crear Director de Proyecto
<b>Nombre de Tarea:</b> Crear Director de proyecto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 09/08/2017	<b>Fecha Fin:</b> 09/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita realizar una inserción en la base de datos de un registro para un director de proyecto, para después asignarle un proyecto de grado el cual va a dirigir.	

**Tabla 127. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU019:</b> Crear Director de Proyecto
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 10/08/2017	<b>Fecha Fin:</b> 10/08/2017
<b>Programador Responsable:</b> Team Development	

**Descripción:**

Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.

**Tabla 128. Task card: Integración**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU019:</b> Crear Director de Proyecto
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 3 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 10/08/2017	<b>Fecha Fin:</b> 10/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).	

**Tabla 129. Historia de usuario Crear Revisión**

<b>Historia de usuario</b>
<b>Número:</b> HU020
<b>Programador Responsable:</b> Team development

<b>Nombre de la historia:</b> Crear Revisión	
<b>Nivel de riesgo de desarrollo:</b> media	
<b>Peso de la historia:</b> 5 puntos de historia	
<b>Tiempo estimado:</b> 19 Horas	
<b>Perspectiva del producto</b>	
<b>Como:</b> Usuario Jurado	
<b>Requiero:</b> Crear una revisión para un proyecto.	
<b>Para:</b> Poder informar al Usuario Estudiante acerca de las correcciones de su proyecto.	
<b>Criterios de aceptación</b>	
<ol style="list-style-type: none"> <li>1. El usuario Jurado debe estar autenticado cuando requiera crear una revisión de un proyecto que debe estar previamente creado y en estado de aceptado.</li> <li>2. Asignar concepto exitoso: Si el usuario Jurado está registrado en la plataforma tendrá la posibilidad de crear una revisión par aun proyecto. Para crear una revisión debe ingresar los siguientes datos: descripción, corrección, selecciona un archivo que desea cargar con la información de la revisión y escoge un proyecto para asociarlo a la revisión, además elige el estado del proyecto. Una vez ingresada toda la información se informa al usuario que la acción se realizó satisfactoriamente.</li> <li>3. Asignar concepto no exitoso: Se envía un mensaje de error informando al usuario que no se puede realizar la operación porque el usuario no ingresó correctamente la información requerida.</li> </ol>	
<b>Actividades</b>	<b>N° de horas de implementación</b>

• Crear interfaces	3
• Conexión al servidor	6
• Crear concepto	5
• Testing	3
• Integración	2

**Tabla 130. Task Card: Crear Interfaces**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea: 1</b>	<b>HU020: Crear Revisión</b>
<b>Nombre de Tarea:</b> Crear Interfaces	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 10/08/2017	<b>Fecha Fin:</b> 10/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear interfaz gráfica para crear revisión: crear etiquetas, campos de texto y botones (Descripción, Corrección, Archivo, Estado, Proyecto).	

**Tabla 131. Task Card: Conexión al servidor**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea: 2</b>	<b>HU020: Crear Revisión</b>

<b>Nombre de Tarea:</b> Conexión al servidor	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 6 Horas	
<b>Fecha Inicio:</b> 11/08/2017	<b>Fecha Fin:</b> 11/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear la conexión entre el servidor y la aplicación web para que responda a las solicitudes realizadas por el usuario en este caso crear revisión.	

**Tabla 132. Task Card: Crear concepto**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 3	<b>HU020:</b> Crear Revisión
<b>Nombre de Tarea:</b> Crear Concepto	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 5 Horas	
<b>Fecha Inicio:</b> 11/08/2017	<b>Fecha Fin:</b> 14/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Crear un método que permita ingresar toda la información que se ha emitido para la creación del concepto.	



**Tabla 133. Task Card: Testing**

<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 4	<b>HU020:</b> Crear Revisión
<b>Nombre de Tarea:</b> Testing	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 3 Horas	
<b>Fecha Inicio:</b> 14/08/2017	<b>Fecha Fin:</b> 14/08/2017
<b>Programador Responsable:</b> Team Development	
<b>Descripción:</b> Realizar el testing de la funcionalidad desarrollada para encontrar errores y/o vulnerabilidades, o darla por terminada.	

**Tabla 134. Task Card: Integración**

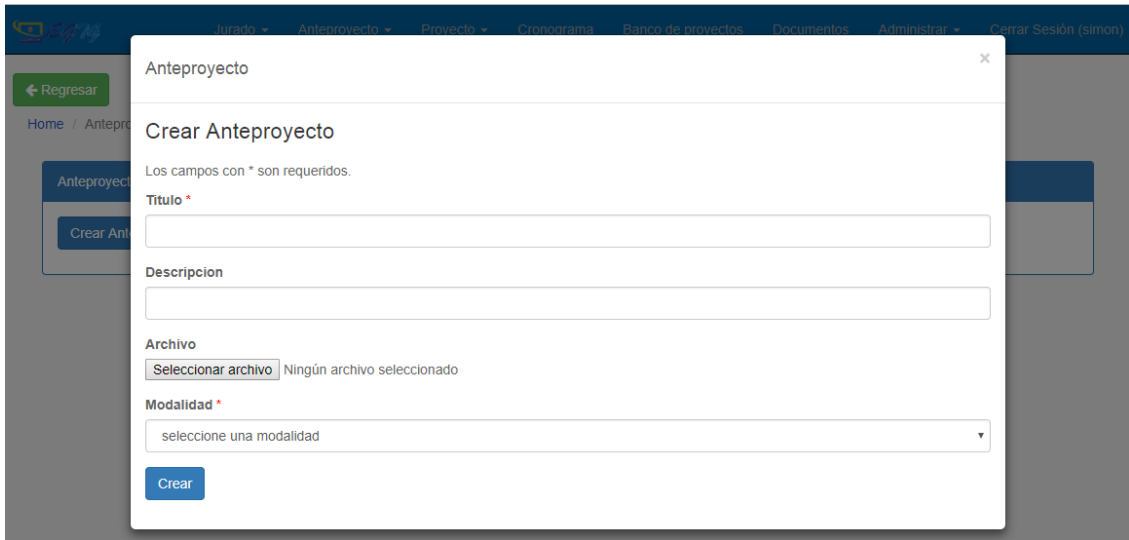
<b>TAREA DE INGENIERÍA</b>	
<b>Número de Tarea:</b> 5	<b>HU020:</b> Crear Revisión
<b>Nombre de Tarea:</b> Integración	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos estimados:</b> 5 puntos de historia
<b>Tiempo de implementación estimado:</b> 2 Horas	
<b>Fecha Inicio:</b> 14/08/2017	<b>Fecha Fin:</b> 14/08/2017

**Programador Responsable:** Team Development

**Descripción:**

Ensamblar el producto a partir de sus componentes y asegurar que una vez integrado se comporta correctamente (funcionalidad y atributos de calidad requeridos).

**4.4.2 Diseño.**



**Figura 26. Interfaz Crear Anteproyecto. [Fuente Propia]**

**4.4.3 Desarrollo.**

**Tabla 135. Evento Programado Crear Anteproyecto.**

<b>EVENTO PROGRAMADO CREAR ANTEPROYECTO</b>
<b>RESULTADOS DE CREAR ANTEPROYECTO</b>

Anteproyecto

Crear Anteproyecto

Los campos con \* son requeridos.

**Titulo \***

Descripcion

**Archivo**

Seleccionar archivo Ningún archivo seleccionado

**Modalidad \***

seleccione una modalidad

Crear

## EVENTOS PROGRAMADOS

```

217     {
218         $model = new Anteproyecto();
219
220         if ($model->load(Yii::$app->request->post())) {
221             // se guarda el id del usuario
222
223             $model->date_create = date('Y-m-d H:m:s');
224             $model->id = Yii::$app->user->id;
225
226             // se obtiene la instancia del archivo subido
227
228             $archivo = $model->nombre;
229             $model->file = UploadedFile::getInstance($model, 'file');
230             $model->file->saveAs('anteproyecto/'.$archivo.'.'.$model->file->extensio
231
232             // guardando el camino en la Bd columna
233
234             $model->archivo_anteproyecto = 'anteproyecto/'.$archivo.'.'.$model->fi
235
236             $model->save(false);
237             return $this->redirect(['view', 'id' => $model->idanteproyecto]);
238         } else {
239             return $this->renderAjax('create', [
240                 'model' => $model,
241             ]);
242         }
243     }
244
245

```

### 4.4.4 Pruebas.

**Tabla 136. Pruebas funcionales Crear Anteproyecto.**

<b>Historia de Usuario</b>	Crear Anteproyecto.
<b>Propósito</b>	Crear un Anteproyecto.
<b>Requisitos</b>	Ser estudiante.

<p><b>Pasos</b></p>	<p>El estudiante ingresa al menú en la pestaña de Anteproyecto, ahí debe seleccionar la opción Crear Anteproyectos, en ese momento aparecerá una lista con los detalles más relevantes del anteproyecto luego se desplegará un formulario con estos campos; título, descripción y además debe seleccionar el documento a subir y la modalidad de sus trabajo de grado. Finalmente se presiona el botón crear.</p>
<p><b>Resultados Esperados</b></p>	<p>Crea el anteproyecto y es enviado al secretario.</p>

**Figura 27. Interfaz Crear anteproyecto.[Fuente Propia]**

#	Estudiante	Titulo	Descripcion	Modalidad	
1	juan	prueba	prueba	trabajo de Investigacion	

**Figura 28. Interfaz Resultado Crear Anteproyecto. [Fuente Propia]**

#### 4.4.5 Sprint Review.

Se entregó funcionalidades, crear jurado, recuperar contraseña, crear anteproyecto, crear proyecto, asignar concepto, crear director proyecto, crear revisión, completamente finalizadas, es decir, software funcional

#### 4.4.6 Retrospectiva.

Se hizo uso de la técnica de la estrella de mar, ya que permite tener retroalimentación, y acorde a esto se ajusta la ejecución de actividades para el siguiente sprint.

**Tabla 137. Retrospectiva del Sprint 3.**

<b>Comenzar a Hacer.</b> <ul style="list-style-type: none"><li>• Pruebas finales de la plataforma web.</li></ul>	<b>Más de.</b> <ul style="list-style-type: none"><li>• Socializar y retroalimentar el proceso.</li><li>• Reuniones del equipo.</li></ul>
<b>Seguir Haciendo.</b> <ul style="list-style-type: none"><li>• Reuniones con el Cliente.</li><li>• Uso de tecnologías para comunicación del equipo.</li></ul>	<b>Menos de.</b> <ul style="list-style-type: none"><li>• Cambios innecesarios a la base de datos.</li></ul>
<b>Dejar de Hacer.</b> <ul style="list-style-type: none"><li>• Cambios en la funcionalidad de la plataforma.</li></ul>	

## 5. RESULTADOS

Con la implementación de la plataforma web, se brinda a los involucrados en el proceso de gestión y seguimiento de los trabajos de grado en la facultad de ingeniería, la automatización de estos mismos.

A continuación se detallan las mejoras, por cada uno de los involucrados:

- **Secretario de la Facultad:** Manejaba el proceso de gestión y seguimiento de los trabajos de grado de la facultad de ingeniería de manera por medio de un formato en Excel, en el cual éste encargado había designado colores diferentes para saber en qué etapa de madurez se encontraba cada uno de los trabajos, y posteriormente de acuerdo a esto se enviaba el concepto correspondiente al estudiante, por ésta razón el proceso era más confuso y necesitaba de más tiempo. Por lo tanto se han agregado funcionalidades en las cuales visualiza todos los anteproyectos, proyectos; ver y asignar jurados, asignar una fecha de sustentación a los proyectos aprobados, ver y crear nuevos eventos en el calendario; cargar y eliminar documentos importantes durante el proceso de los trabajos de grado; y además recibir alertas (correos electrónicos) para estar al tanto de los tiempos que se deben manejar en cada entrega por parte de los involucrados en el proceso del trabajo de grado, esto con el fin de optimizar tiempos de entrega durante el proceso.

El secretario de la facultad puede generar reportes, para tener un control de la información.

Además puede descargar formatos y documentos pertinentes durante el proceso.

- **Estudiante:** Manejaba el proceso de gestión y seguimiento de su trabajo de grado de una manera que le permitía tomarse el tiempo que quisiera para hacer sus respectivas entregas de informes, o también se olvidaba del tiempo que le restaba para realizar dichas entregas por ello el proceso se hacía un poco más largo. Por lo

tanto se adicionaron las funcionalidades de recibir alertas (correo electrónico) las cuales indican cuantos días le restan para hacer sus respectivas entregas, optimizando tiempos en el proceso del trabajo de grado.

Además el estudiante tendrá un contador de días, el cual empezará en 365 a partir de la fecha en que se matricule e irá disminuyendo diariamente.

Los estudiantes que en su defecto aún no se encuentran matriculados en trabajo de grado, podrán ingresar en la pestaña de Banco de proyectos, en la cual se encuentran algunos proyectos para escoger y realizarlos.

Además puede descargar documentos importantes, y formatos que debe diligenciar en la Universidad.

- **Miembro del Comité:** Manejaba el proceso de gestión y seguimiento de su trabajo de grado de una manera que le permitía tomarse el tiempo que quisiera para entregar su concepto del anteproyecto, o también se olvidaba del tiempo que le restaba para realizar dichas entregas por ello el proceso se hacía un poco más largo. Por lo tanto se adicionaron las funcionalidades de recibir alertas (correo electrónico) las cuales indican cuantos días le restan para hacer sus respectivas entregas de conceptos, optimizando los tiempos en el proceso del trabajo de grado.
- **Docente:** Se adicionó la funcionalidad de Banco de Proyectos, que permite a un docente agregar proyectos nuevos, para que el secretario los analice, los publique y posteriormente ser escogidos por algún estudiante.
- **Jurado:** Manejaba el proceso de gestión y seguimiento de su trabajo de grado de una manera que le permitía tomarse el tiempo que quisiera para entregar su concepto del proyecto, o también se olvidaba del tiempo que le restaba para realizar dichas entregas por ello el proceso se hacía un poco más largo. Por lo tanto se adicionaron las funcionalidades de recibir alertas (correo

electrónico) las cuales indican cuantos días le restan para hacer sus respectivas entregas de conceptos.

Además podrá descargar los respectivos documentos importantes, y formatos que debe diligenciar posteriormente en la Universidad.

Finalmente, con el desarrollo de la plataforma web se logra optimizar los tiempos de entregas durante el proceso de trabajos de grado.



## 6. CONCLUSIONES

La facultad de ingeniería de la Corporación Universitaria Autónoma del Cauca hacía uso de una herramienta ofimática, que generaba pérdida de tiempo y duplicación de información durante el proceso de trabajos de grado, además de que no existía un sistema que alertara oportunamente a los usuarios para cumplir con los tiempos establecidos en la resolución 0047, para realizar las entregas de su respectiva documentación. El manejo manual de los datos que arroja la gestión de información de los trabajos de grado genera pérdida de información y retrasos en tiempos, y por otra parte hay que tener en cuenta que el día que haya cambio de personal, los procesos serán difíciles de aprender pues no están definidos, sino que al contrario los maneja unas pocas personas y solo los conocen ellos, por estas razones surge la idea de automatizar los procesos de trabajos de grado de la facultad de ingenierías.

Para el desarrollo del proyecto se utilizó la metodología de trabajo Extreme Programming combinado con las prácticas de scrum, que ayuda a que el trabajo se haga en equipo y haya avances continuos, además permite la entrega oportuna de un prototipo escalable y funcional basado en las fases de planificación, diseño, desarrollo y pruebas de la metodología utilizada: Extreme Programming.

La plataforma web brinda una gestión de la información organizada y de más fácil acceso, además tiene como valor agregado un sistema de alarmas que avisa a sus usuarios los tiempos exactos de entregas y recepción de trabajos y correcciones, cumpliendo de manera efectiva la norma 0047 que exige el cumplimiento de estos tiempos y brindando a la facultad una herramienta a nivel organizativo y de gestión que ayuda en la toma de decisiones.

Concluyendo de esta manera que la plataforma web organiza la información, permite a los usuarios un fácil seguimiento de sus trabajos, brinda la oportunidad de un banco de proyectos, notifica a cada usuario tiempos y tareas pendientes, logrando así tiempo de respuestas óptimos,

con fácil acceso a la información y un nivel organizacional alto, teniendo en cuenta el fácil uso de esta.

### **6.1 TRABAJO FUTURO.**

En este proyecto de investigación se automatizaron los procesos referentes a trabajos de grado de la facultad de ingenierías de la Corporación Universitaria Autónoma del Cauca, como trabajo futuro se podrá implementar el sistema en las otras facultades de la universidad, puesto que la plataforma se puede adaptar fácilmente a los procesos utilizados en las otras dependencias. Se le podrán agregar nuevas funcionalidades que se necesiten de acuerdo a las necesidades de cada facultad.

Por otro lado, como modelo de negocio es poder brindarles a las universidades de la región caucana la oportunidad de utilizar la plataforma puesto que es notable la ausencia de aplicativos que gestionen los proyectos de grado.

## 7. BIBLIOGRAFÍA

- [1] “Orange Scrum para Colombia y Latino América”. [Online]. Disponible en: <http://tecnoweb2.com/taxonomy/term/303> . Consultado el 19 de Noviembre de 2017
- [2]”Orange Scrum para la gestión de Proyectos”. [Online]. Disponible en: <http://tecnoweb2.com/software-gestion-de-proyectos-orangescrum-colombia-latinoamerica> . Consultado el 19 de Noviembre de 2017
- [3] P.L. Torres, E.A. Sánchez López (2003 noviembre 12) “Metodologías Ágiles en el desarrollo de software” [online]. Disponible en: <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
- [4] “Metodologías Ágiles en el desarrollo de software”. [Online]. Disponible en: <http://grupo1u6.blogspot.com.co/2016/03/beneficios-de-aplicar-metodologias.html> . Consultado el 29 de Enero de 2017.
- [5] A. Navarra Cadavid, J.D. Fernández Martínez, J. Morales Vélez (2013) “Revisión de metodologías ágiles para el desarrollo de software” [online]. Disponible en: <https://dialnet.unirioja.es/descarga/articulo/4752083.pdf>
- [6] “Metodologías Ágiles para el desarrollo de software: eXtreme programming (xp)”. <http://www.cyta.com.ar/ta0502/v5n2a1.htm> . Consultado el 23 de Octubre de 2016.
- [7] M. Bermejo. “El Kanban”. [Online]. Disponible en: [https://www.exabyteinformatica.com/uoc/Audiovisual/Produccion\\_multimedia/Produccion\\_multimedia\\_\(Modulo\\_4\).pdf](https://www.exabyteinformatica.com/uoc/Audiovisual/Produccion_multimedia/Produccion_multimedia_(Modulo_4).pdf)
- [8] M.Palomo. Duarte, I. Montero Pérez. “Programación en PHP a través de ejemplos”. [Online]. Disponible en: [http://servicio.uca.es/softwarelibre/publicaciones/apuntes\\_php](http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php)
- [9] <https://hipertextual.com/archivo/2013/05/entendiendo-html5-guia-para-principiantes/> . Consultado el 01 de noviembre de 2016.

- [10] A. Enríquez Toleda, J. Maldonado Ayala, Y. Nakamura Ortega, G. Noguérón Toledo. "MySQL". [Online]. Disponible en: <http://www.gridmorelos.uaem.mx/~mcruz/cursos/miic/MySQL.pdf>
- [11] J.J. Gutiérrez. "¿Qué es un framework web?". [Online]. Disponible en: [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf)
- [12] "Yii framework". [Online]. Disponible en: <http://www.yiiframework.com/doc/guide/1.1/es/quickstart.what-is-yii>
- [13] <https://www.ecured.cu/CSS3> . Consultado el 20 de marzo de 2017.
- [14] K. Garimella, M. Lees, B. Williams "BPM (GERENCIA DE PROCESOS DE NEGOCIO)". [Online]. Disponible en: [http://www.konradlorenz.edu.co/images/publicaciones/suma\\_digital\\_sistemas/bpm.pdf](http://www.konradlorenz.edu.co/images/publicaciones/suma_digital_sistemas/bpm.pdf)
- [15] <http://dro2754.blogspot.com.co/2010/03/diferencia-entre-plataforma-y-pagina.html> Consultado el 14 de octubre de 2017
- [16] L. Gimson. (2012 junio). "Metodologías ágiles y desarrollo basado en conocimiento". [Online]. Disponible en: [http://sedici.unlp.edu.ar/bitstream/handle/10915/24942/Documento\\_completo\\_.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/24942/Documento_completo_.pdf?sequence=1)
- [17] <https://ingsoftwarekarlacevallos.wordpress.com/2015/05/08/metodologia-de-desarrollo-agil-xp-y-scrum/>. Consultado el 21 de marzo de 2017
- [18] "Reglas y Prácticas en extreme Programming". [Online]. Disponible en: <https://ie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>. Consultado el 21 de marzo de 2017.
- [19] "Fases de la programación extrema". [Online]. Disponible en: <http://programacionextrema.tripod.com/fases.htm> . Consultado el 23 de Octubre de 2016
- [20] S.M. Meléndez Valladarez, M.E. Gaitán, N.N Pérez Reyes (2016 enero 28). "Metodología Ágil Programación Extrema XP". [Online]. Disponible en: <http://repositorio.unan.edu.ni/1365/1/62161.pdf>

- [21] <http://www.pruebasdesoftware.com/pruebadeaceptacion.htm> . Consultado el 21 de marzo de 2017
- [22] N. Juristo, A.M. Moreno, S. Vegas (2006 octubre 17) "Técnicas de Evaluación de Software" [online]. Disponible en: [http://www.grise.upm.es/htdocs/sites/extras/12/pdf/Documentacion\\_Evaluacion\\_7.pdf](http://www.grise.upm.es/htdocs/sites/extras/12/pdf/Documentacion_Evaluacion_7.pdf)
- [23] L. Calabria, P. Piriz. (2003) "Metodología XP". [Online]. Disponible en: [http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia\\_xp.pdf](http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia_xp.pdf)
- [24] "Desarrollo Ágil de software-Caso Programación Extrema-XP". [Online]. Disponible en : <http://ldc.usb.ve/~abianc/materias/ci4713/metodologiasagiles.pdf> . Consultado el 24 de Octubre de 2016
- [25] Procesossoftware. "Metodología Ágil XP". [Online]. Disponible en: <https://procesossoftware.wikispaces.com/M%C3%A9todo+%C3%81gil+XP>. Consultado el 24 de Octubre de 2016.
- [26] <https://proyectosagiles.org/que-es-scrum/>. Consultado el 21 de marzo de 2017
- [27] Mountain Goat Software "Scrum". [Online]. Disponible en: <http://www.mountaingoatsoftware.com/agile/scrum>
- [28] "Agile software development through Scrum". [Online]. Disponible en: <https://springtimesoft.co.nz/agile-software-development-scrum/>. Consultado el 24 de Octubre de 2016.
- [29] J. Palacio. "Gestión de proyectos Scrum Manager" (2014 Abril) versión 2.5. [Online]. Disponible en: [http://www.scrummanager.net/files/sm\\_proyecto.pdf](http://www.scrummanager.net/files/sm_proyecto.pdf)
- [30] <https://www.linkedin.com/pulse/how-adopt-scrum-agile-within-your-marketing-team-mark-appel>. Consultado el 23 de febrero de 2017

- [31] K. Schwaber, J. Sutherland (2013 Julio). “La guía de Scrum”. [Online]. Disponible en: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- [32] “Facilitador (Scrum Master)”. [Online]. Disponible en: <https://proyectosagiles.org/facilitador-scrum-master/> . Consultado el 24 de Octubre de 2016
- [33] <https://cedaniel200.blogspot.com.co/2014/04/scrum.html>. Consultado el 23 de febrero de 2017.
- [34] Scrum Manager Body of knowledge. “Artefactos”. [Online]. Disponible en: <http://www.scrummanager.net/bok/index.php?title=Artefactos>
- [35] <http://www.becominganagilearchitect.com/scrum-eventos-el-sprint>. Consultado el 23 de febrero de 2017.
- [36] <http://jmbeas.es/guias/agile-inception/>. Consultado el 17 de enero de 2017.
- [37] H. Kniberg. “Scrum y XP desde las trincheras”. [Online]. Disponible en: <http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf>
- [38] <https://proyectosagiles.org/demostracion-requisitos-sprint-review/> . Consultado el 24 de octubre de 2016.
- [39] Cadena Crítica. “Revisión del Sprint- Sprint Review”. [Online]. Disponible en: <http://www.cadenacritica.com/revision-del-sprint-sprint-review/>
- [40] <https://proyectosagiles.org/retrospectiva-sprint-retrospective/> . Consultado el 25 de Octubre de 2016
- [41] <https://proyectosagiles.org/replanificacion-proyecto/> Consultado el 01 de noviembre de 2016.
- [42] <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. Consultado el 28 de marzo de 2017

- [43] <http://tutorialesenpdf.com/xampp/>. Consultado el 28 de marzo de 2017.
- [44] <http://web.tursos.com/como-conectar-php-a-mysql-en-un-servidor-local/>. Consultado el 28 de marzo de 2017.
- [45] <https://www.phpmyadmin.net/>. Consultado el 28 de marzo de 2017.
- [46] <http://www.peam.es/cursos/free-learning/servidor-web/instalar-phpmyadmin.php>. Consultado el 28 de marzo de 2017.
- [47] Andrea, (2014 mayo 23) "Github y Bitbucket: servicios de Git en la nube". [Online]. Disponible en: <https://hipertextual.com/archivo/2014/05/github-y-bitbucket/>
- [48] <http://dandovueeltasconmisequipos.blogspot.com.co/2015/10/git-bitbucket-sourcetree.html>. Consultado el 28 de marzo de 2017
- [49] [http://librosweb.es/libro/javascript/capitulo\\_1.html](http://librosweb.es/libro/javascript/capitulo_1.html). Consultado el 28 de marzo de 2017
- [50] C. Ferreira, (2015 septiembre 6), "¿Qué es bootstrap? Bootstrap es un framework front end". [Online]. Disponible en: <http://www.negocioscaninos.com/que-es-bootstrap-bootstrap-framework-front-end/>
- [51] <http://www.mexired.com/blog/que-es-jquery>. Consultado el 28 de marzo de 2017
- [52] [http://programacion.net/articulo/uso\\_de\\_la\\_clase\\_phpmailer\\_213](http://programacion.net/articulo/uso_de_la_clase_phpmailer_213). Consultado el 28 de marzo de 2017
- [53] <http://culturacion.com/que-es-apache/>. Consultado el 28 de marzo de 2017
- [54] <https://proyectosagiles.org/2009/07/01/estimacion-planificacion-agil-quinto-encuentro-agil-barcelona/>. Consultado el 28 de marzo de 2017
- [55] R. de la Torre. (2014 octubre 15). "Estimación de tareas en Scrum"- [Online]. Disponible en: <http://blog.opensistemas.com/estimacion-de-tareas-en-scrum/> consultado el 28 de marzo de 2017

[56] “Atom”. [Online]. Disponible en: <https://atom.io/>. Consultado el 28 de marzo de 2017

[57] <https://franguerrero.blog.wordpress.com/2015/10/11/editores-de-codigo-para-desarrollo-web/> . Consultado el 28 de marzo de 2017

[58] <http://www.wordreference.com/definicion/gesti%C3%B3n> Consultado el 17 de marzo de 2017.

[59] Y. Sánchez Castro. (2010 septiembre 17) <http://ingesoft2010.blogspot.com.co/2010/09/retroalimentacion.html>. [Online]. Disponible en: Consultado el 17 de marzo de 2017.

[60] <http://deconceptos.com/ciencias-sociales/test> . Consultado el 17 de marzo de 2017.

[61] [http://epf.eclipse.org/wikis/openupsp/openup\\_basic/guidances/concepts/iteration,\\_lam4ADkBEduxovfWMDsntw.html](http://epf.eclipse.org/wikis/openupsp/openup_basic/guidances/concepts/iteration,_lam4ADkBEduxovfWMDsntw.html) . Consultado el 17 de marzo de 2017.

[62] <https://www.ecured.cu/Framework> . Consultado el 17 de marzo de 2017.

[63] <http://conceptodefinicion.de/metodologia/> . Consultado el 17 de marzo de 2017.

[64] J. Alonso Nocelo (2015 junio 9). “La Retrospectiva: Estrella de mar”. [Online]. Disponible en: <http://www.elartedepresentar.com/2015/06/la-retrospectiva-estrella-de-mar/>. Consultado el 23 de marzo de 2017

[65] Miguel (2015 enero 20). “Introducción a la arquitectura cliente-servidor (1)”. [Online]. Disponible en: <http://mialtoweb.es/introduccion-a-la-arquitectura-cliente-servidor-1/> . Consultado el 23 de marzo de 2017.