

**REINGENIERÍA DE LA APLICACIÓN UISPORT QUE SOPORTA LA  
ENSEÑANZA DE PROCESOS DE REPRESENTACIÓN EN INGENIERÍA DEL  
SOFTWARE**

**MARIA FERNANDA FUENTES MANTILLA  
VIVIANA PAOLA TORRES GUERRERO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FISICO-MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA  
BUCARAMANGA  
2014**

**REINGENIERÍA DE LA APLICACIÓN UISPORT QUE SOPORTA LA ENSEÑANZA DE  
PROCESOS DE REPRESENTACIÓN EN INGENIERÍA DEL SOFTWARE**

**MARIA FERNANDA FUENTES MANTILLA  
VIVIANA PAOLA TORRES GUERRERO**

**Trabajo de grado para optar por el título de Ingeniero de Sistemas**

**DIRECTORA  
M.Sc. NAYIBE DEL CARMEN RUIZ CHAGÜI**

**CODIRECTORA  
PhD. SONIA CRISTINA GAMBOA SARMIENTO**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FISICO-MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA  
BUCARAMANGA**

**2014**

## **DEDICATORIA**

*A Dios por ser el mejor guía en este camino.*

*A mi familia en general, por creer en mí y por su apoyo incondicional. En especial a mi madre y mi abuela quienes durante todo este proceso han sido mi inspiración y ejemplo para optar por el título universitario.*

*A mis amigos y compañeros que hicieron parte de esta meta haciendo que mi paso por la universidad fuera una de las mejores e inolvidables experiencias en mi vida.*

**PAOLA TORRES**

## DEDICATORIA

*A Dios por guiar mis pasos y permitirme culminar esta etapa de mi vida.*

*A mis padres por su apoyo incondicional. Especialmente a mi madre que ha estado a mi lado siempre luchando por mis sueños y ayudándome a cumplirlos.*

*A toda mi familia por estar presente en cada momento de mi vida. Especialmente a mi tía Nubia y a Erika Andrea por siempre creer en mí.*

*A mis abuelos que desde el cielo me cuidan y me acompañan.*

*A mi novio Fabio Andrés por ser mi compañero y amigo en este camino, y por su apoyo constante con el cual logre cumplir este sueño.*

*A mi compañera y amiga Viviana por apoyarme siempre a pesar de los obstáculos que se nos presentaron.*

*A mis mejores amigos que han hecho parte de este sueño y que dejaron en mí recuerdos y momentos inolvidables.*

**MARIA FERNANDA FUENTES M.**

## **AGRADECIMIENTOS**

A Dios por ser nuestro guía en este proceso y por darnos sabiduría y mucha fortaleza para culminar esta etapa de nuestras vidas satisfactoriamente.

A nuestros padres que incondicionalmente han estado a nuestro lado para apoyarnos y guiarnos en todas las etapas de nuestra vida, siendo así también merecedores de este logro, pues sin ellos nada de esto hubiese sido posible.

A nuestros seres queridos, familiares y amigos que nos acompañaron día a día en este proceso y nos apoyaron e hicieron parte de nuestra formación como personas.

A las profesoras Nayibe del Carmen Ruiz Chagüi y Sonia Cristina Gamboa Sarmiento por su apoyo, sus enseñanzas y por ser nuestras guías en el desarrollo de este proyecto. También por confiar en nosotras y permitirnos crecer no solo profesional sino personalmente.

A la Universidad Industrial de Santander y a la Escuela de Ingeniería de Sistemas e Informática por la formación profesional recibida y por permitirnos realizar este proyecto.

Y a todos aquellos que de alguna forma hicieron parte de todo nuestro camino por la Universidad, y creyeron en nosotras.

## TABLA DE CONTENIDO

<b>INTRODUCCIÓN</b>	<b>19</b>
<b>1. DESCRIPCIÓN DEL PROYECTO</b>	<b>21</b>
1.1 PLANTEAMIENTO Y JUSTIFICACION DEL PROBLEMA	21
1.2 OBJETIVOS	22
1.2.1 <i>Objetivo general</i>	22
1.2.2 <i>Objetivos específicos</i>	22
<b>2. MARCO DE REFERENCIA</b>	<b>23</b>
2.1 ANTECEDENTES DEL PROYECTO	23
2.1.1 <i>Portafolio digital – Uisport</i>	23
2.2 FUNDAMENTOS TEÓRICOS	23
2.3 REINGENIERÍA	24
2.3.1 <i>Análisis de inventarios</i>	25
2.3.2 <i>Reestructuración de documentos</i>	25
2.3.3 <i>Ingeniería inversa</i>	25
2.3.4 <i>Reestructuración</i>	28
2.3.5 <i>Ingeniería hacia adelante</i>	29
2.4 E-LEARNING	29
2.5 ENTORNO VIRTUAL DE APRENDIZAJE	30
2.6 HERRAMIENTAS TECNOLÓGICAS	31
2.6.1 <i>Moodle</i>	31
2.6.2 <i>PHP (Hypertext Pre-Processor)</i>	32
2.6.3 <i>MYSQL</i>	32
2.7 PRUEBAS DE SOFTWARE	33
2.7.1 <i>Objetivos de las pruebas de software</i>	34
2.7.2 <i>Estrategias de las pruebas</i>	34
2.7.3 <i>Pruebas de aceptación</i>	36
2.7.4 <i>Técnicas de las pruebas</i>	36
2.8 MANTENIMIENTO DE SOFTWARE	38
2.8.1 <i>Tipos de mantenimiento de software</i>	38
2.9 CALIDAD DE SOFTWARE	38
<b>3. METODOLOGIA</b>	<b>40</b>

3.1 ANÁLISIS DE INVENTARIOS _____	40
3.2 REESTRUCTURACIÓN DE DOCUMENTOS _____	40
3.3 INGENIERÍA INVERSA _____	41
3.3.1 <i>Ingeniería inversa para comprender datos</i> _____	41
3.3.2 <i>Ingeniería inversa para entender el procesamiento</i> _____	41
3.3.3 <i>Ingeniería inversa de interfaces de usuario</i> _____	42
3.4 REESTRUCTURACIÓN _____	42
3.5 INGENIERÍA HACIA ADELANTE _____	42
3.6 SOFTWARE PRUEBAS DE FASE INICIAL _____	43
3.6.1 <i>Pruebas de funcionalidad</i> _____	43
3.7 MEJORAS REALIZADAS _____	50
<b>4. DESCRIPCIÓN DE LA NUEVA HERRAMIENTA _____</b>	<b>74</b>
<b>5. PRUEBA PILOTO _____</b>	<b>78</b>
5.1 ANALISIS DE LOS RESULTADOS _____	81
<b>6. CONCLUSIONES _____</b>	<b>92</b>
<b>7. RECOMENDACIONES _____</b>	<b>94</b>
<b>BIBLIOGRAFIA _____</b>	<b>95</b>
<b>ANEXOS _____</b>	<b>97</b>

## LISTA DE FIGURAS

FIGURA 1. MODELO DE PROCESO DE REINGENIERÍA DE SOFTWARE	24
FIGURA 2. PROCESO DE INGENIERÍA INVERSA	26
FIGURA 3. ESTRATEGIAS DE LAS PRUEBAS DE SOFTWARE	34
FIGURA 4. PRUEBAS CAJA NEGRA	37
FIGURA 5. PESTAÑA OBTENCIÓN DE REQUERIMIENTOS VERSIÓN 1.0	50
FIGURA 6. PESTAÑA OBTENCIÓN DE REQUERIMIENTOS VERSIÓN 2.0	51
FIGURA 7. PESTAÑA DISEÑO	51
FIGURA 8. BOTONES GUARDAR CAMBIOS Y VOLVER AL MENÚ	52
FIGURA 9. ADVERTENCIA DE ABANDONO DE PÁGINA	52
FIGURA 10. CUADRO DE TEXTO DESCONFIGURADO	53
FIGURA 11. CUADRO DE TEXTO NUEVA VERSIÓN	53
FIGURA 12. MENÚ DESPLEGABLE VERSIÓN 1.0	54
FIGURA 13. ENUNCIADO DEL PROBLEMA	54
FIGURA 14. FUNCIÓN EXPANDIR	54
FIGURA 15. “POR DEFINIR” VERSIÓN 1.0	55
FIGURA 16. “POR DEFINIR” VERSIÓN 2.0	56
FIGURA 17. EXPANDIR VERSIÓN 1.0	56
FIGURA 18. EXPANDIR VERSIÓN 2.0	57
FIGURA 19. MENÚ ANÁLISIS VERSIÓN 1.0	57
FIGURA 20. MENÚ ANÁLISIS VERSIÓN 2.0	58
FIGURA 21. BOTONES GENERAR Y AYUDA DEL SISTEMA VERSIÓN 1.0	59
FIGURA 22. BOTONES GENERAR Y AYUDA DEL SISTEMA VERSIÓN 2.0	59
FIGURA 23. HEURÍSTICO VERSIÓN 1.0	60
FIGURA 24. HEURÍSTICO VERSIÓN 2.0	60
FIGURA 25. CARACTERES ESPECIALES EN EL PDF VERSIÓN 1.0	61
FIGURA 26. CARACTERES ESPECIALES EL EN PDF VERSIÓN 2.0	61
FIGURA 27. FECHA Y HORA EN EL PDF VERSIÓN 1.0	62
FIGURA 28. FECHA Y HORA EN EL PDF VERSIÓN 2.0	62
FIGURA 29. OPCIONES EN EL CUADRO DE TEXTO VERSIÓN 1.0	63
FIGURA 30. RESULTADO DEL CUADRO DE TEXTO EN EL PDF VERSIÓN 1.0	63
FIGURA 31. OPCIONES EN EL CUADRO DE TEXTO VERSIÓN 2.0	64
FIGURA 32. CASOS DE USO VERSIÓN 1.0	64
FIGURA 33. TABLA CASOS DE USO VERSIÓN 2.0	65
FIGURA 34. ERROR EN PESTAÑA INFORMACIÓN	65
FIGURA 35. PESTAÑA INFORMACIÓN CORREGIDA	66
FIGURA 36. BOTÓN VOLVER AL MENÚ	66

<b>FIGURA 37. ICONOS Y TOOL TIPS VERSIÓN 1.0</b>	<b>67</b>
<b>FIGURA 38. ICONOS Y TOOL TIPS VERSIÓN 2.0</b>	<b>67</b>
<b>FIGURA 39. ICONO COMPARTIR</b>	<b>67</b>
<b>FIGURA 40. OPCIÓN DESCARGAR VERSIÓN 1.0</b>	<b>68</b>
<b>FIGURA 41. OPCIÓN DESCARGAR VERSIÓN 2.0</b>	<b>68</b>
<b>FIGURA 42. ROLES VERSIÓN 1.0</b>	<b>69</b>
<b>FIGURA 43. ROLES VERSIÓN 2.0</b>	<b>69</b>
<b>FIGURA 44. BOTÓN DIAGRAMA DE CASO DE USO</b>	<b>70</b>
<b>FIGURA 45. MENÚ EVIDENCIAS VERSIÓN 1.0</b>	<b>70</b>
<b>FIGURA 46. MENÚ PORTAFOLIOS COMPARTIDOS VERSIÓN 1.0</b>	<b>70</b>
<b>FIGURA 47. MENÚ EVIDENCIAS VERSIÓN 2.0</b>	<b>71</b>
<b>FIGURA 48. MENÚ PORTAFOLIOS COMPARTIDOS VERSIÓN 2.0</b>	<b>71</b>
<b>FIGURA 49. BOTÓN SIGUIENTE ÍTEM</b>	<b>71</b>
<b>FIGURA 50. BOTÓN EDITAR DESDE ANÁLISIS</b>	<b>72</b>
<b>FIGURA 51. BOTÓN SEGUIR EDITANDO</b>	<b>72</b>
<b>FIGURA 52. BOTÓN NUEVO CASO DE USO</b>	<b>72</b>
<b>FIGURA 53. DESCRIPCIÓN PANTALLA PRINCIPAL</b>	<b>76</b>
<b>FIGURA 54. AYUDA DEL SISTEMA</b>	<b>76</b>
<b>FIGURA 55. PDF GENERADO</b>	<b>77</b>
<b>FIGURA 56. SUBIR EVIDENCIA</b>	<b>77</b>
<b>FIGURA 57. ESTUDIANTES ENTRANDO A LA PLATAFORMA</b>	<b>79</b>
<b>FIGURA 58. ESTUDIANTES CREANDO UN PROYECTO EN LA PLATAFORMA</b>	<b>79</b>
<b>FIGURA 59. ESTUDIANTES LLENANDO LA ENCUESTA</b>	<b>80</b>
<b>FIGURA 60. ANÁLISIS PREGUNTA 1</b>	<b>81</b>
<b>FIGURA 61. ANÁLISIS PREGUNTA 2</b>	<b>82</b>
<b>FIGURA 62. ANÁLISIS PREGUNTA 3</b>	<b>83</b>
<b>FIGURA 63. ANÁLISIS PREGUNTA 4</b>	<b>84</b>
<b>FIGURA 64. ANÁLISIS PREGUNTA 5</b>	<b>85</b>
<b>FIGURA 65. ANÁLISIS PREGUNTA 6</b>	<b>86</b>
<b>FIGURA 66. ANÁLISIS PREGUNTA 7</b>	<b>87</b>
<b>FIGURA 67. ANÁLISIS PREGUNTA 8</b>	<b>88</b>
<b>FIGURA 68. ANÁLISIS PREGUNTA 9</b>	<b>89</b>
<b>FIGURA 69. ANÁLISIS PREGUNTA 10</b>	<b>90</b>
<b>FIGURA 70. ANÁLISIS GENERAL DE LA PRUEBA PILOTO</b>	<b>91</b>
<b>FIGURA 71. BASE DE DATOS</b>	<b>97</b>
<b>FIGURA 72. CASO DE USO: GENERAL</b>	<b>98</b>
<b>FIGURA 73. CASO DE USO: OBTENCION Y ANALISIS</b>	<b>99</b>
<b>FIGURA 74. CASO DE USO: DEFINIR CASO DE USO Y DEFINIR ESCENARIO</b>	<b>99</b>
<b>FIGURA 75. CASO DE USO: DISEÑO</b>	<b>100</b>

<b>FIGURA 76. CASO DE USO: EVIDENCIA</b>	<b>100</b>
<b>FIGURA 77. CASO DE USO: VISTA</b>	<b>101</b>
<b>FIGURA 78. CASO DE USO: DOCENTE</b>	<b>101</b>
<b>FIGURA 79. BLOQUE PARA DOCENTE</b>	<b>110</b>
<b>FIGURA 80. BLOQUE PARA ESTUDIANTE.</b>	<b>110</b>
<b>FIGURA 81. DESCRIPCIÓN PANTALLA PRINCIPAL</b>	<b>111</b>
<b>FIGURA 82. INFORMACIÓN</b>	<b>112</b>
<b>FIGURA 83. PESTAÑA OBTENCIÓN DE REQUERIMIENTOS</b>	<b>112</b>
<b>FIGURA 84. HEURÍSTICO OBTENCIÓN DE REQUERIMIENTOS</b>	<b>113</b>
<b>FIGURA 85. PDF</b>	<b>113</b>
<b>FIGURA 86. MENÚ OBTENCIÓN DE REQUERIMIENTOS</b>	<b>114</b>
<b>FIGURA 87. OPCIONES EDITAR-VOLVER</b>	<b>114</b>
<b>FIGURA 88. MODELOS</b>	<b>115</b>
<b>FIGURA 89. ESCENARIOS</b>	<b>115</b>
<b>FIGURA 90. ESCENARIO GUARDADO-OPCIONES</b>	<b>116</b>
<b>FIGURA 91. INGRESAR CASO DE USO</b>	<b>116</b>
<b>FIGURA 92. GUARDAR CASO DE USO-OPCIONES</b>	<b>117</b>
<b>FIGURA 93. DIAGRAMAS/MODELOS</b>	<b>117</b>
<b>FIGURA 94. ANÁLISIS DE REQUERIMIENTOS</b>	<b>118</b>
<b>FIGURA 95. HEURÍSTICO ANÁLISIS DE REQUERIMIENTOS</b>	<b>118</b>
<b>FIGURA 96. OPCIÓN EDITAR- ANÁLISIS DE REQUERIMIENTOS</b>	<b>119</b>
<b>FIGURA 97. DISEÑO</b>	<b>119</b>
<b>FIGURA 98. HEURÍSTICO DISEÑO</b>	<b>120</b>
<b>FIGURA 99. MENÚ DISEÑO</b>	<b>120</b>
<b>FIGURA 100. EVIDENCIAS</b>	<b>121</b>
<b>FIGURA 101. NUEVA EVIDENCIA</b>	<b>121</b>
<b>FIGURA 102. SUBIR UNA EVIDENCIA</b>	<b>122</b>
<b>FIGURA 103. EVIDENCIAS-OPCIONES</b>	<b>122</b>
<b>FIGURA 104. COMPARTIR EVIDENCIA DESDE VISTAS</b>	<b>123</b>
<b>FIGURA 105. INGRESAR VISTA</b>	<b>124</b>
<b>FIGURA 106. COMPARTIR EVIDENCIA-USUARIOS</b>	<b>124</b>
<b>FIGURA 107. VISTAS-OPCIONES</b>	<b>125</b>
<b>FIGURA 108. PORTAFOLIOS COMPARTIDOS</b>	<b>125</b>
<b>FIGURA 109. COMENTARIOS EN LA EVIDENCIA COMPARTIDA</b>	<b>126</b>
<b>FIGURA 110. DESCARGAR PORTAFOLIO COMPARTIDO</b>	<b>126</b>
<b>FIGURA 112. ACTIVAR EDICIÓN</b>	<b>130</b>
<b>FIGURA 113. AGREGAR BLOQUE</b>	<b>131</b>
<b>FIGURA 114. E-PORTAFOLIO</b>	<b>131</b>
<b>FIGURA 115. BLOQUE E-PORTAFOLIO</b>	<b>132</b>

## LISTA DE TABLAS

<b>TABLA 1. PRUEBAS DE FUNCIONALIDAD</b>	<b>45</b>
<b>TABLA 2. CASO DE USO: OBTENCION Y ANALISIS</b>	<b>102</b>
<b>TABLA 3. CASO DE USO: DEFINIR CASO DE USO</b>	<b>103</b>
<b>TABLA 4. CASO DE USO: DISEÑO</b>	<b>104</b>
<b>TABLA 5. CASO DE USO: EVIDENCIA</b>	<b>105</b>
<b>TABLA 6. CASO DE USO: VISTA</b>	<b>105</b>
<b>TABLA 7. CASO DE USO: DOCENTE</b>	<b>106</b>
<b>TABLA 8. ARCHIVOS MODIFICADOS</b>	<b>128</b>
<b>TABLA 9. CARPETAS</b>	<b>129</b>
<b>TABLA 10. ENCUESTA PRUEBA PILOTO</b>	<b>133</b>

## LISTA DE ANEXOS

<b>ANEXO A. BASE DE DATOS</b>	<b>97</b>
<b>ANEXO B. MANUAL DE USUARIO</b>	<b>108</b>
<b>ANEXO C. MANUAL TECNICO</b>	<b>127</b>
<b>ANEXO D. BLOQUE E-PORTAFOLIO</b>	<b>130</b>
<b>ANEXO E. PRUEBA PILOTO: HERRAMIENTA UISPORT 2.0</b>	<b>133</b>

## RESUMEN

**TÍTULO:** REINGENIERÍA DE LA APLICACIÓN UISPORT QUE SOPORTA LA ENSEÑANZA DE PROCESOS DE REPRESENTACIÓN EN INGENIERÍA DEL SOFTWARE<sup>\*</sup>.

**AUTORES:** María Fernanda Fuentes Mantilla, Viviana Paola Torres Guerrero<sup>\*\*</sup>

### **PALABRAS CLAVES**

TIC, reingeniería, portafolio digital, ingeniería del software, estrategia didáctica.

### **DESCRIPCIÓN**

Las nuevas tecnologías de la Información y la Comunicación (TIC) ofrecen día a día una extensa gama de oportunidades y posibilidades en el proceso de enseñanza y aprendizaje, permitiendo romper las barreras del tiempo y el espacio. Es durante este proceso que nace el *e-learning*, el cual permite la creación de ambientes virtuales de aprendizaje en donde se establezca una red de información permanente y una constante comunicación profesor-estudiante que incluye la creación de contenidos, y la administración de los procesos de aprendizaje, evaluación y realimentación.

*Uisport* es una estrategia didáctica que surgió como el resultado de una investigación que pretendía la creación de una herramienta que apoyara el aprendizaje del concepto de *representación* en procesos de diseño de software y lo desarrollara como una habilidad en los estudiantes. Dicha herramienta se llevó a cabo durante la asignatura Ingeniería del Software I del programa de Ingeniería de Sistemas e Informática en el año 2012, que ya en uso evidenció inconvenientes de usabilidad en interacción con el usuario; esto conllevó a realizar una reingeniería y allí nació Uisport 2.0.

La reingeniería responde a la evolución de la tecnología y la informática, pues promete un cambio verdaderamente rápido y significativo para mantenerse a tono con el cambiante mundo de las TIC.

Este proyecto de grado fue desarrollado con el objetivo de brindar al estudiante no solo una herramienta para el estudio y el desarrollo de software, sino un ambiente orientado al trabajo en equipo y al aprendizaje flexible, donde se invitan a desarrollar las habilidades necesarias para modelar la realidad.

---

\* Trabajo de grado

\*\* Facultad de Ingeniería Físico - Mecánicas, Escuela de Ingeniería de Sistemas e Informática.

## ABSTRACT

**TITLE:** REENGINEERING THE APPLICATION UISPORT THAT SUPPORTS THE TEACHING OF REPRESENTATION IN SOFTWARE ENGINEERING PROCESSES<sup>\*</sup>.

**AUTHORS:** María Fernanda Fuentes Mantilla, Viviana Paola Torres Guerrero <sup>\*\*</sup>

### KEYWORDS

TIC, reengineering, digital portfolio, software engineering, teaching strategy.

### DESCRIPTION

New technologies of information and communication technology (TIC) offer every day a wide range of opportunities and possibilities in the process of teaching and learning, enabling you to break down the barriers of time and space. It is during this process that is born e-learning, which allows the creation of virtual learning environments where establishes a network of permanent information and constant communication teacher-student that includes content creation, and administration of the processes of learning, evaluation and feedback.

Uisport is a teaching strategy that emerged as the result of an investigation aimed at the creation of a tool that will support the learning of the concept of representation in software design processes and develop it as a skill in students. This tool was carried out during the subject Software Engineering I of the systems engineering and computer science program in 2012, in use showed drawbacks of usability in interaction with the user; This pitting make a reengineering and there emerged Uisport 2.0.

Reengineering responds to the evolution of technology and computer science, for it promises a truly rapid and significant change to keep in tune with the changing world of TIC.

This graduation project was developed with the objective of providing the student not only a tool for the study and development of software, but an environment oriented to team work and flexible learning, where they invite you to develop the skills necessary to model reality.

---

\* Bachelor thesis.

\*\* Facultad de Ingeniería Físico - Mecánicas, Escuela de Ingeniería de Sistemas e Informática.

## INTRODUCCIÓN

La reingeniería responde a la evolución de la tecnología y la informática, pues promete un cambio verdaderamente rápido y significativo para mantenerse a tono con el cambiante mundo de las nuevas tecnologías de la Información y la Comunicación.

El presente proyecto muestra el proceso de reingeniería aplicado a la herramienta *Uisport* que se viene utilizando a partir del segundo semestre del 2012 por los estudiantes de pregrado de Ingeniería de sistemas e informática, como apoyo a la asignatura de Ingeniería de software I.

El capítulo 1 nos presenta la descripción, la justificación y los objetivos bajo los cuales se desarrolló este proyecto de grado.

En el segundo capítulo de este documento se exponen los fundamentos teóricos bajo los cuales se desarrolló el proyecto. En primer lugar se encuentra la descripción de la primera versión de la herramienta, la metodología, las herramientas pedagógicas y tecnológicas, las pruebas de software implementadas, los tipos de pruebas, el mantenimiento de software y por último la calidad del mismo.

El trabajo realizado, fue evaluado mediante el modelo cíclico del proceso de reingeniería, propuesto por Pressman en su libro *Ingeniería del Software, Un enfoque práctico*, el cual hizo posible verificar la funcionalidad de la herramienta *Uisport* y mejorar los servicios ofrecidos con el fin de que se adapte a las nuevas exigencias tecnológicas que surgen diariamente. El capítulo 3 de este documento cuenta paso a paso la implementación de la metodología, las pruebas iniciales realizadas al software y las fallas encontradas junto con las mejoras implementadas.

*Uisport 2.0* busca seguir apoyando el aprendizaje virtual, enfocado en el concepto de representación en procesos de desarrollo de software no solo como conocimiento sino también como una habilidad a ser desarrollada por el estudiante, con el fin de convertirlo en un excelente profesional. La nueva herramienta cuenta con mejoras que pueden ser apreciadas en la descripción de la misma en el capítulo 4.

La finalidad de este proyecto es verificar y refinar las funcionalidades de *Uisport 1.0*, y mejorar y optimizar el software con énfasis en la usabilidad y la educación virtual. Es por esto que para dar por terminado el proyecto, se realizó una prueba piloto con los estudiantes de la asignatura Ingeniería del software I, del segundo semestre del 2014, descrita en el capítulo 5. A través de esta prueba los estudiantes pudieron conocer las

diferentes funcionalidades de la herramienta y dieron a conocer su punto de vista como usuarios finales.

Para terminar con este proceso, se hizo un análisis general contemplado en los capítulos 6 y 7, donde se presentan las conclusiones y recomendaciones que han surgido de la realización de este proyecto.

## 1. DESCRIPCIÓN DEL PROYECTO

El presente proyecto titulado: **“REINGENIERÍA DE LA APLICACIÓN UISPORT QUE SOPORTA LA ENSEÑANZA DE PROCESOS DE REPRESENTACIÓN EN INGENIERÍA DEL SOFTWARE”** se desarrolló con el fin de mejorar la herramienta *Uisport 1.0*, que se puso a prueba en la asignatura Ingeniería del Software I del programa de pregrado Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander a partir del año 2012, como resultado de una investigación de maestría.

Este proyecto está dirigido a los estudiantes de la asignatura ya mencionada, con el fin de que desarrollen un conjunto de habilidades de modelado de la realidad, que respondan a requerimientos determinados.

El objetivo del capítulo está enfocado en ubicar al lector en el por qué y el cómo se realizó este trabajo, mencionando su justificación, sus objetivos y la metodología que se implementó para el cumplimiento de los requerimientos definidos.

Gran parte del trabajo realizado, fue evaluado mediante el modelo cíclico del proceso de reingeniería, el cual hizo posible verificar la funcionalidad de la herramienta *Uisport* y mejorar los servicios ofrecidos con el fin de que se adapte a las nuevas exigencias tecnológicas que surgen diariamente.

La intervención en la reingeniería de la estrategia didáctica *Uisport*, estuvo basada en las diferentes pruebas software que se le realizaron no solo para determinar aquellos procesos que se debían mejorar, sino también para verificar que el usuario estuviera a gusto con la nueva versión, y que se estaba cumpliendo con los requerimientos establecidos para el software.

### 1.1 PLANTEAMIENTO Y JUSTIFICACION DEL PROBLEMA

El ejercicio de la ingeniería consiste primordialmente en diseñar; en el caso de la ingeniería de sistemas este diseño se orienta a la concepción de sistemas de información que contribuyan a la solución de problemáticas del entorno. Tales soluciones parten de la representación que en el ejercicio profesional, se hagan los ingenieros, tanto de la situación problemática como de la solución. Así, los estudiantes de ingeniería de sistemas deben desarrollar habilidades de representación para llevar a cabo procesos de desarrollo de software que se requieran.

Por su parte, el mercado ofrece herramientas informáticas que permiten hacer modelado a partir de los estándares de representación que ofrece la notación UML, pero su uso eficiente no depende sólo de sus funcionalidades, sino que se requiere de conocimientos y habilidades para plantear modelos que representen adecuadamente tanto las situaciones problemáticas como las soluciones propuestas.

En ese contexto se diseñó y se desarrolló una aplicación para la enseñanza de procesos propios de la ingeniería del software en la que se asumió el portafolio como estrategia didáctica; pero esta herramienta, ya en el uso con los estudiantes, evidenció la necesidad de refinar los procesos mismos a partir de los cuales se hizo su diseño y su desarrollo.

Es necesario, entonces, rediseñar tales procesos y llevar a cabo el desarrollo de software correspondiente para que la aplicación se ajuste a las necesidades pedagógicas y tecnológicas de los usuarios.

## **1.2 OBJETIVOS**

### **1.2.1 Objetivo general**

Desarrollar la versión 2.0 de la aplicación *Uisport* para la plataforma Moodle mediante un proceso de reingeniería.

### **1.2.2 Objetivos específicos**

- Evaluar la herramienta *Uisport 1.0*, mediante pruebas de software con el fin de determinar las mejoras a realizar y requerimientos a tener en cuenta en la nueva versión.
- Rediseñar los procesos que soportan la estrategia didáctica portafolio, documentarlos e implementarlos con el fin de mejorar la calidad de la aplicación original sin que pierda su funcionalidad.
- Desarrollar pruebas software y prueba piloto con el fin de validar los requerimientos del nuevo sistema.

## 2. MARCO DE REFERENCIA

### 2.1 ANTECEDENTES DEL PROYECTO

A través de la realización de este proyecto de grado, se busca evaluar la funcionalidad y los requerimientos de la versión 1.0 de la herramienta *Uisport*, para conocer las partes del software que serán sometidas a reingeniería, y así mejorar ciertos aspectos. Para ello es importante conocer las características de *Uisport*, aspectos técnicos y su funcionalidad.

Por otra parte, se muestran algunas herramientas similares que han sido punto de referencia.

#### 2.1.1 Portafolio digital – *Uisport*

*Uisport* es un dispositivo computacional que apoya los procesos de enseñanza y aprendizaje del curso “ingeniería del software i”, del programa de pregrado de ingeniería de sistemas e informática.

Su estrategia didáctica se basa en la herramienta *portafolios* y hace uso de principios pedagógicos y los heurísticos planteados por Bruegge y Dutoit para el desarrollo de software orientado a objetos.

*Uisport* se implementó con la herramienta de programación libre php y conexión a bases de datos con *MySQL*. Fue construido para la versión 2.4.1 de Moodle2, a manera de bloque para poder ser incorporado dentro de un curso.

Se encuentra alojado en un *hosting* público que hace parte del grupo de investigación en Sistemas y Tecnologías de la Información (STI), y fue utilizado el primer semestre del 2012.

### 2.2 FUNDAMENTOS TEÓRICOS

El desarrollo del software no se detiene cuando un sistema se entrega, sino que continúa a lo largo de la vida de este, con el fin de mantenerlo útil. Los cambios en las expectativas de los usuarios generan nuevos requerimientos para el software existente. Otro motivo son las compañías que trabajan con software y que están basadas en antiguas reglas empresariales, ya que cuando deciden modificarlas, el software también debe modificarse y adaptarse a ellas.

Michael Hammer<sup>1</sup> en un artículo para *Harvard Business Review* propone una revolución en los procesos empresariales y de computación:

---

<sup>1</sup> (Hammer, Reengineer Work: Don't Automate, Obliterate, 1990)

”La reingeniería lucha por liberarse de las antiguas reglas acerca de cómo organizamos y dirigimos nuestros negocios”

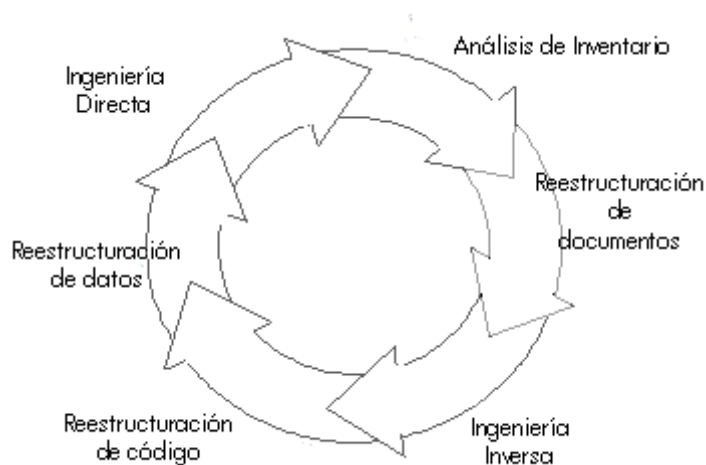
Como toda revolución, esta trajo cambios tanto positivos como negativos. Por ello, las compañías que deciden actualizar las reglas empresariales que las rigen, deben modificar o reconstruir partes del software para adaptarse a los cambios y mejorar su rendimiento.

### 2.3 REINGENIERÍA

Durante años, las compañías en su afán por atender las necesidades empresariales, se dedicaron a adaptar y corregir la aplicación con la que contaban, dejando a un lado las buenas prácticas de ingeniería; logrando serios efectos colaterales. Todos estos problemas de mantenimiento de software, llevaron a la búsqueda de una solución que hoy llamamos reingeniería.

La reingeniería es un proceso que toma tiempo, dinero y recursos; y que por ello es necesario que las compañías tengan una estrategia establecida, para implementarla. Es una actividad de reconstrucción para la cual se tiene un modelo de proceso de reingeniería del software que defina las actividades a realizar y los principios a seguir.

**Figura 1.** Modelo de proceso de reingeniería de software



Fuente: (Pressman, 2010)

La figura 1. Según Pressman, representa un *modelo cíclico*, es decir, una cadena de ciclos auto sostenibles. Esto significa que las actividades allí presentadas pueden ser

revisadas y que el proceso puede terminar después de cualquiera de estas. A continuación se describe cada una de las actividades: (Pressman, 2010)

### **2.3.1 Análisis de inventarios**

En toda compañía u organización de software debe contarse con el inventario de las aplicaciones que se encuentran activas, y con información sobre cada aplicación. Al organizar toda esta información según algunos criterios, se encuentran las aplicaciones a las cuales se les hará reingeniería.

Este inventario debe realizarse regularmente, debido a que el estado de las aplicaciones cambia con el tiempo, y con ello cambian las prioridades para aplicar reingeniería.

### **2.3.2 Reestructuración de documentos**

La mayoría de los sistemas heredados cuentan con una documentación débil. Para estos casos, existen tres opciones:

- Si un programa funciona, es relativamente estático y se acerca el final de su vida útil, la solución es dejarlo así y no invertir tiempo en la creación de la documentación; siempre y cuando las posibilidades de que presente grandes cambios sean casi nulas.
- Cuando las organizaciones cuentan con pocos recursos, se adopta el enfoque “documento cuando toque”. En lugar de documentar por completo una aplicación, solo se documentaran las porciones del sistema que requieren cambios.
- Incluso cuando las aplicaciones requieren de una documentación completa, se sugiere que solo se realiza tanta documentación como sea necesaria para entender el software.

### **2.3.3 Ingeniería inversa**

La ingeniería inversa para software es el proceso de analizar un programa para representarlo en un nivel superior de abstracción que el código fuente. Es un proceso de recuperación de diseño, que usa herramientas que extraen información de diseño de datos, arquitectónico y procedimental de un programa existente.

La ingeniería inversa obtiene esta información por medio del código fuente, pero el nivel de abstracción, la completitud de la documentación, la relación entre las herramientas y el analista humano, y la direccionalidad del proceso pueden variar constantemente.

El *nivel de abstracción*, debe ser tan alto como sea posible. Ser capaz de inferir representaciones de diseño procedimental (bajo nivel), información de estructura de

programa y datos (nivel medio), modelos de objeto, datos y flujo de control (relativamente alto), y modelos de relación de entidad (nivel alto). Si aumenta el nivel, se facilita la comprensión del programa.

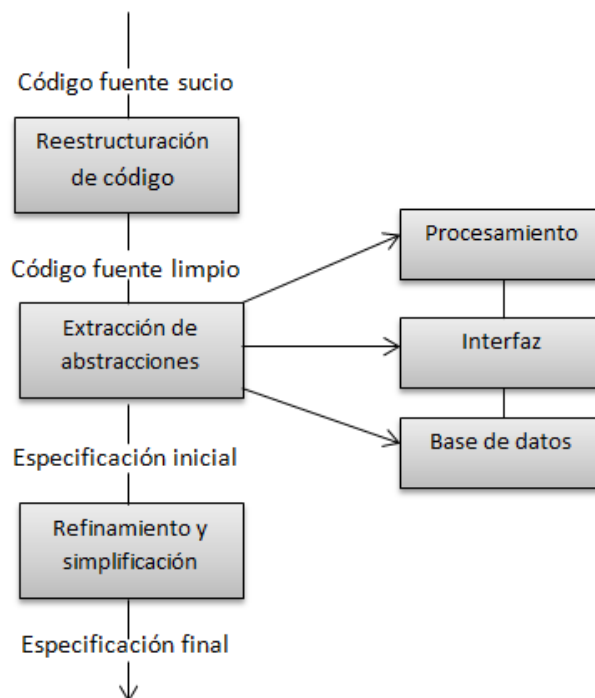
La *completitud* se refiere al nivel de detalle que se proporciona en un nivel de abstracción. Generalmente, la completitud disminuye cuando el nivel de abstracción aumenta.

La completitud mejora en proporción directa a la cantidad de análisis realizado durante la ingeniería inversa.

La *interactividad* se refiere a la integración del ser humano con las herramientas automatizadas. Generalmente la interactividad aumenta con el nivel de abstracción, o la completitud decaerá.

Si la *direccionalidad* de la ingeniería inversa es de una vía, toda la información obtenida del código fuente, pasa al ingeniero de software que la usara en cualquier mantenimiento. Si es de dos vías, toda la información pasa a una herramienta de reingeniería que regenerara el programa antiguo.

**Figura 2.** Proceso de ingeniería inversa



Fuente: (Pressman, 2010)

En la figura 2. Se observa el proceso de ingeniería inversa, el cual antes de poder comenzar, según lo menciona Pressman, debe reestructurar el código fuente sucio para que sea más fácil de leer, ya que es la base para todas las actividades de ingeniería inversa posteriores.

El núcleo de este proceso es la *extracción de abstracciones*. Se debe evaluar el programa antiguo y a partir del código fuente, especificar el proceso que se realiza de la interfaz de usuario que se aplica y de las estructuras de datos del programa, o la base de datos que se usa.

### **Ingeniería inversa para comprender datos:**

Esta ocurre en diferentes niveles de abstracción y generalmente es la primera tarea de reingeniería. En el nivel de programa, las estructuras de datos internas deben someterse a ingeniería inversa para una reingeniería global. En el nivel del sistema, las estructuras de datos globales se someten a reingeniería para implementar nuevos paradigmas de administración de bases de datos.

**Estructuras de datos internas.** Para este tipo de datos, la ingeniería inversa se enfoca en la definición de clases de objetos. Se examina el código, para agrupar variables relacionadas. En muchos casos se identifican tipos de datos abstractos.

**Estructura de la base de datos.** La reingeniería en una base de datos requiere comprender los datos existentes y sus relaciones.

### **Ingeniería inversa para entender el procesamiento:**

Se comienza por comprender y extraer abstracciones representadas en el código, para lo cual se analiza este en varios niveles de abstracción: sistema, programa, componente, patrón y enunciado.

Es importante conocer la funcionalidad global del sistema antes de iniciar la ingeniería inversa más detallada. Esto informa sobre los conflictos de interoperabilidad entre aplicaciones del sistema. Cada uno de los programas del sistema de aplicación representa una abstracción funcional en un nivel alto de detalle; y para representar la interacción entre estas abstracciones, se crea un diagrama de bloques. Para cada componente se desarrolla una narrativa de procesamiento. Cuando se considera el código dentro de un componente, se debe buscar secciones de código que representen patrones procedimentales genéricos. En la mayoría de los componentes, una sección de código prepara los datos para procesamiento, otra lo realiza y otra

prepara los resultados para exportarlos. Dentro de cada sección se encuentran patrones más pequeños.

Para sistemas grandes, la ingeniería inversa tiene un enfoque semiautomatizado. Se usan herramientas para la comprensión de la semántica del código. De aquí se pasa a la reestructuración de herramientas de ingeniería hacia adelante.

#### **Ingeniería inversa de interfaces de usuario :**

Antes de poder reconstruir una interfaz de usuario, debe realizarse ingeniería inversa.

Para comprender una interfaz de usuario existente, se deben especificar la estructura y el comportamiento de la interfaz, así:

- Las acciones básicas que debe procesar la interfaz
- La descripción de la respuesta del sistema a dichas acciones
- La relevancia de la equivalencia de interfaces

La notación de modelado de comportamiento aporta información a los dos primeros ítems. La información necesaria para crear un modelo de comportamiento, surge de la manifestación externa de la interfaz existente. Si se necesita información adicional debe extraerse del código.

#### **2.3.4 Reestructuración**

La reestructuración de software modifica el código fuente y/o los datos para hacerlos sensibles a cambios futuros.

No suele modificar la arquitectura global del sistema, si no que se enfoca en detalles de diseño de módulos individuales y estructuras de datos locales. Ocurre cuando la arquitectura básica es sólida, y se inicia cuando grandes partes del software son aprovechables y solo un subconjunto de módulos necesita cambios extensos.

#### **Reestructuración de código:**

Se realiza con el fin de producir un diseño con la misma funcionalidad pero con mejor calidad que el programa original. Generalmente las técnicas usadas modelan la lógica del programa usando algebra booleana y aplicando unas reglas de transformación que producen lógica reestructurada.

#### **Reestructuración de datos:**

Antes de comenzar, debe realizarse un tipo de ingeniería inversa llamada *análisis de código fuente*. Se evalúan todos los enunciados de lenguaje de programación para

extraer ítems de datos y objetos, obtener información sobre flujo de datos y entender las estructuras de datos existentes, este proceso es conocido como análisis de datos. Después se comienza el rediseño de datos. La estandarización de registros de datos clarifica las definiciones de los datos para lograr consistencia entre los nombres de ítems dentro de un formato. Otra forma de rediseño, llamada racionalización de nombre de datos, garantiza que las convenciones de nomenclatura sigan estándares locales, y que los sobrenombres se eliminen. Cuando este proceso avanza, se realizan modificaciones físicas a las estructuras de datos existentes para que el diseño sea más efectivo.

### **Ingeniería hacia adelante**

Cuando se cuenta con un programa mal diseñado, y poco documentado, se tienen las siguientes opciones:

- Implementar los cambios necesarios combatiendo el diseño ad hoc y el código fuente enredado.
- Comprender los funcionamientos interiores del programa, para hacer modificaciones más efectivas.
- Rediseñar, recodificar y poner a prueba porciones de software que requieran modificación, aplicando un enfoque de ingeniería de software.
- Rediseñar, recodificar y poner a prueba el programa, y usar la reingeniería para comprender el diseño actual.

Si el desarrollador interno de software con una gran producción de programas, se deben clasificar por importancia y decidir cuales aplican para ingeniería hacia adelante.

Este proceso de ingeniería hacia adelante usa el enfoque de ingeniería de software para volver a crear una aplicación existente. En la mayoría de los casos, se logra un programa con más capacidades que la antigua aplicación.

### **2.4 E-LEARNING**

Los cambios logrados en los hábitos de estudio y de trabajo por TIC, han repercutido en todo tipo de organizaciones. Los modelos tradicionales de enseñanza han cambiado considerablemente gracias a que las TIC ofrecen nuevas posibilidades superando las limitaciones del espacio y del tiempo.

El *e-learning* surge como un espacio en el que se aprovecha la Internet para proveer un amplio despliegue de soluciones que mejoren el conocimiento, y que proporcione

información permanente, comunicación, desarrollo de nuevas competencias y el uso de herramientas informáticas en estrategias didácticas.

Algunos de los beneficios que ofrece esta herramienta son (Rosenberg, 2002, págs. 30,31) :

- Permite que muchos usuarios aprendan y compartan información simultáneamente.
- Permite actualizar o modificar la información y mantener al tanto a los usuarios en tiempo real.
- Se tiene acceso a la información en cualquier momento del día.
- Permite que todos los usuarios reciban virtualmente el mismo material, sin importar el tipo de plataforma que se use.
- Incentiva el trabajo en equipo y la creación de comunidades de práctica, gracias a las herramientas de comunicación entre los usuarios.
- Cuenta con elementos didácticos como imágenes, videos, textos, animaciones, que incentivan al usuario a continuar usando esta herramienta.

El uso de *e-learning* no solo nos abre las puertas hacia un mundo de tecnología y nuevas formas de aprender, si no también nos presenta una nueva forma de pensar acerca de cómo aprendemos. En la actualidad las personas pueden aprender de muchas formas, mediante el acceso a información bien diseñada, por el uso de herramientas que mejoran el desempeño a través de la experiencia y de otros factores.

## **2.5 ENTORNO VIRTUAL DE APRENDIZAJE**

Conocido como VLE por sus siglas en inglés (Virtual Learning Environment), o LMS (Learning Management System), o Sistema Gestor de Aprendizaje o Ambiente Virtual de Aprendizaje (AVA), es un espacio en el que se realizan actividades relacionadas con los procesos de enseñanza y aprendizaje.

En un VLE los espacios didácticos no requieren de espacio físico, ni de la presencia de los usuarios, debido a que se aprovecha el uso del internet para realizar este proceso. Tiene la capacidad de que tanto docentes como estudiantes intercambien información, y logren el proceso de aprendizaje de manera dinámica y efectiva.

Un VLE puede ser utilizado como:

- Sistema de distribución de aprendizaje. Este proceso consiste en proporcionar cierto tipo de material por medio del espacio virtual, para complementar las clases

presenciales. Este material incluye todo tipo de información sobre el curso, herramientas de aprendizaje, entre otros materiales didácticos, a los que el usuario o estudiante podrá acceder en cualquier momento de manera remota.

- Sistema de formación. En este caso, el VLE toma el control del proceso central de aprendizaje y enseñanza. Por lo tanto, es allí donde se desarrolla en su totalidad dicho proceso.

## 2.6 HERRAMIENTAS TECNOLÓGICAS

### 2.6.1 Moodle

Moodle es un sistema de código abierto de gestión de contenidos (CMS), también conocido como Sistema de Gestión de Aprendizaje (LMS) o un Entorno Virtual de Aprendizaje (EVA). Moodle es un paquete de software para producir cursos basados en Internet y sitios web. Se trata de un proyecto de desarrollo global diseñado para apoyar un marco de educación social constructivista.

El objetivo del proyecto Moodle es siempre dar a los educadores las mejores herramientas para gestionar y promover el aprendizaje.

Hay muchas maneras de utilizar Moodle:

- Moodle tiene características que le permiten escalar a grandes despliegues y cientos de miles de estudiantes, sin embargo, también se puede utilizar para una escuela primaria.
- Muchas instituciones lo utilizan como su plataforma para realizar cursos totalmente en línea, mientras que algunos lo utilizan simplemente para aumentar los cursos cara a cara (conocida como *blended learning*).
- Muchos usuarios usan los módulos de actividad (por ejemplo, foros, bases de datos y wikis) para construir comunidades más ricas de colaboración de aprendizaje en torno a su objeto (en la tradición constructivista social), mientras que otros prefieren utilizar Moodle como una forma de entregar el contenido a los estudiantes (como paquetes SCORM estándar) y evaluar el aprendizaje mediante tareas o cuestionarios.

Moodle se distribuye gratuitamente como software de código abierto (bajo la Licencia Pública General de GNU). Esto significa que Moodle tiene derechos de autor, pero que tienen libertades adicionales. Usted está autorizado a copiar, usar y modificar Moodle siempre que acepte: proporcionar la fuente a otros, no modificar o eliminar la licencia y derechos de autor original, y aplicar esta misma licencia a cualquier trabajo derivado.

Moodle se puede instalar en cualquier ordenador que pueda ejecutar PHP, y puede soportar una base de datos tipo SQL (por ejemplo, MySQL). Se puede ejecutar en los sistemas operativos Windows, Mac y Linux.

### **2.6.2 PHP (Hypertext Pre-Processor)**

Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se enviara la página a través de internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que vera el cliente. El cliente solo recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

#### **Características**

- Es un lenguaje multiplataforma completamente orientado al desarrollo de páginas web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con *MySQL* y *PostgreSQL*.
- Es libre, por lo que se presenta como una alternativa fácil de acceso para todos.

### **2.6.3 MYSQL**

Mysql es un sistema de administración de bases de datos (database management system, dbms) para bases de datos relacionales. mysql no es más que una aplicación que permite gestionar archivos llamados de bases de datos.

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

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

También es muy destacable, la condición de *open source* de *Mysql*, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, de tal forma que se puede descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de *Mysql* una de las herramientas más utilizadas de los programadores orientados a internet.

## 2.7 PRUEBAS DE SOFTWARE

Las pruebas de software son procesos realizados concurrentemente a través de las etapas del desarrollo de software que utiliza y mantiene el *testware* y cuyo objetivo es apoyar la disminución del riesgo de aparición de fallas y faltas en operación.<sup>2</sup>

Realizar pruebas es uno de los procesos más importantes para la evaluación del software que se está desarrollando, existen diferentes técnicas y estrategias que se llevan a cabo con la intención de descubrir errores. Todo esto ayudara a que el proceso sea más eficiente.

Las pruebas constituyen un proceso muy importante para la calidad del producto software ya que ayudan a encontrar los errores presentes dentro del mismo, lo cual evitara a futuro: perdidas económicas, de personal, de información, etc.; por lo cual las organizaciones desarrolladoras de software están destinando entre 30 y 40 por ciento del esfuerzo de un proyecto a la realización de pruebas.<sup>3</sup>

Las pruebas de software implican tiempo pero esto certifica su objetivo principal que es garantizar la calidad del producto desarrollado, lo que implica una serie de aspectos como lo son:

- Verificar la interacción de componentes.
- Verificar la integración adecuada de los componentes.
- Verificar que todos los requisitos se han implementado correctamente.
- Identificar y asegurar que los defectos encontrados se corrijan antes de entregar el software al cliente.
- Diseñar las pruebas necesarias para encontrar las fallas lo cual no debe implicar mucho tiempo ni esfuerzo.

La prueba es la actividad cuyo objetivo es determinar si una pieza de código (un método, una clase o un programa) produce el comportamiento pretendido.

---

<sup>2</sup> [www.acis.org.co](http://www.acis.org.co). recuperado el 31 de julio de 2014, de [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/XXVII\\_Salon\\_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXVII_Salon_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf).

<sup>3</sup> Pressman Roger. Ingeniería del software-Un enfoque práctico. Op.cit. p.218

### 2.7.1 Objetivos de las pruebas de software

En el cuadro de las pruebas de software se encuentran muchos objetivos dentro de los cuales podemos considerar los siguientes como los más importantes:<sup>4 5</sup>

3. Para mostrar al desarrollador y al cliente que el software satisface sus requerimientos.
4. Para descubrir errores en el software en el que el comportamiento de este es incorrecto o, no deseable o no cumple su especificación.
5. Lograr confianza acerca del nivel de calidad
6. Proveer información
7. Prevenir defectos

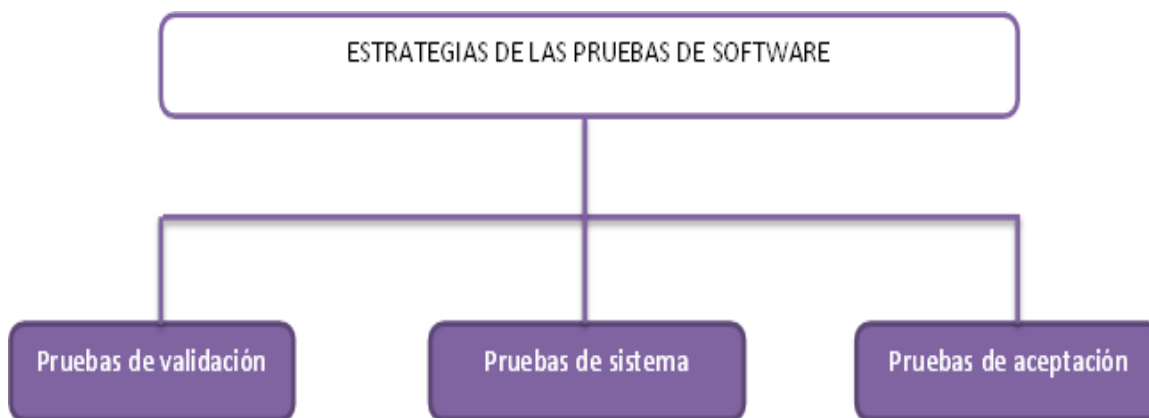
### 2.7.2 Estrategias de las pruebas

Como el autor Pressman<sup>6</sup> lo indica en su libro “una estrategia de prueba del software integra las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software.”

Las estrategias de las pruebas constituyen una parte muy importante en el proceso del desarrollo del software por lo tanto se deben planificar, seguir y realizar los pasos que correspondan dependiendo de la estrategia elegida y así ejecutarlas y agruparlas para la debida evaluación de los datos obtenidos.

En el desarrollo de este proyecto se consideraron las siguientes estrategias de las pruebas de software.

**Figura 3.** Estrategias de las pruebas de software



<sup>4</sup> Sommerville Ian. Ingeniería del software. Séptima Edición. Madrid Pearson Educación 2005. P.492.

<sup>5</sup> [www.acis.org.co](http://www.acis.org.co). Op.cit.

<sup>6</sup> Pressman Roger. Ingeniería del software-Un enfoque práctico. Op. cit. p. 305.

## Pruebas de validación

Cuando el producto software esta ensamblado correctamente y en su totalidad se hace necesario hacer una prueba más, la prueba de validación en la cual se prueban todos los módulos en conjunto para así determinar si se están cumpliendo con todos los requerimientos. En esta práctica existen dos tipos: la prueba alfa y la prueba beta.

En la prueba alfa el software se utiliza en un entorno natural mientras el desarrollador está al tanto para registrar los errores y problemas de uso, estas pruebas son realizadas en entornos controlados. Mientras que las pruebas beta se aplican n un lugar de trabajo de los usuarios finales y a diferencia de la alfa, por lo general el desarrollador no está, por lo tanto el usuario es el que registrar los errores que el aprecia en la aplicación.<sup>7</sup>

Existen algunos criterios para este tipo de pruebas:<sup>8</sup>

La validación del software de logra mediante una serie de pruebas que demuestren que se cumplen los requisitos.

Un plan de prueba delinea la clase de pruebas que se aplicaran y un procedimiento de prueba define los casos de prueba específicos.

Después de que se ha dirigido cada caso de prueba de validación, existirá dos condiciones posibles: 1. La característica de funcionamiento o desempeño cumple con la especificación. 2. Se descubre una desviación de la especificación y se crea una lista de deficiencias.

- **Pruebas de sistema:** Las pruebas del sistema son aquellas que se hacen cuando ya se tienen todos los módulos conectados entre sí, a que a pesar de que puedan estar funcionando bien individualmente, puede que en conjuntos estén ocasionando muchos errores que no se han podido detectar. En el marco de las pruebas de sistema se pueden destacar cuatro tipos:<sup>9</sup>
- **Prueba de recuperación:** esta prueba lo que busca es hacer que el sistema falle de varias maneras y verifica que su recuperación se realice apropiadamente.
- **Prueba de seguridad:** se centra en la comprobación del funcionamiento apropiado de los mecanismos de control al momento de sufrir interrupciones inapropiadas.

---

<sup>7</sup> Ibid. p. 316.

<sup>8</sup> Ibid. p.316.

<sup>9</sup> Ibid. p. 317-318.

- **Prueba de resistencia:** ejecuta el sistema de tal forma que los recursos que demande de los mismos en cantidad, volumen o frecuencia, sean anormales, para así poder tener una idea de cuánto resiste el sistema.
- **Prueba de desempeño:** está diseñada principalmente para probar el desempeño del software en tiempo de ejecución dentro de contexto de un sistema integrado.

### 2.7.3 Pruebas de aceptación

Estas son las últimas pruebas que se hacen y consiste en que el usuario interactúe con el software dentro de su propio entorno de trabajo, para que así él pueda dar su visto bueno de acuerdo con la satisfacción de los requerimientos establecidos previamente.

### 2.7.4 Técnicas de las pruebas

#### ➤ Pruebas funcionales:

- **Prueba de caja blanca**

Las pruebas de caja blanca buscan un examen del detalle procedimental. En esta técnica se prueban las rutas lógicas, los bucles, condiciones y el estado del programa en diferentes puntos para asegurarse de que la operación interna del software se ajuste a las especificaciones.<sup>10</sup> Además permiten diseñar casos de prueba para examinar la lógica del programa que garanticen que:

- Se ejercitan todos los caminos independientes de cada módulo.
- Se ejercitan todas las decisiones lógicas.
- Se ejecutan todos los bucles
- Se ejecutan las estructuras de datos internas.

- **Prueba de caja negra**

Las pruebas de caja negra dejan a un lado la parte lógica del sistema y se centra en la interfaz, en los resultados de las entradas es decir, se centra en los requisitos funcionales del software. Es a su vez un enfoque complementario de las pruebas de caja blanca. Dentro de esta técnica se puede encontrar algunos tipos de prueba como: métodos gráficos, partición equivalente, análisis de valores límite, prueba de tabla ortogonal.<sup>11</sup>

Las pruebas funcionales o de caja negra son una política de selección de casos de pruebas basado en la especificación del componente o programa

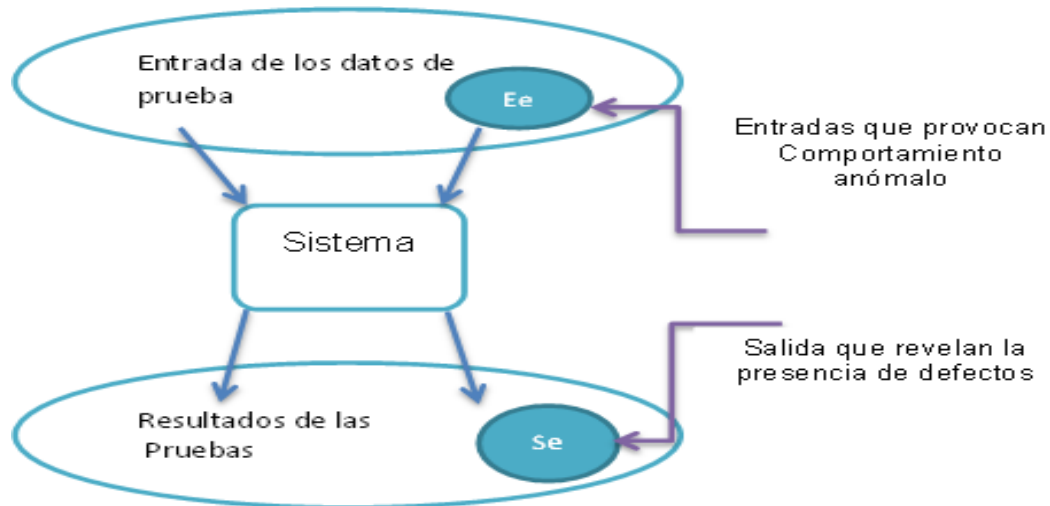
---

<sup>10</sup> Ibid. p. 286-294.

<sup>11</sup> Ibid. p. 294-299.

Las pruebas se seleccionan en función de la especificación y no de la estructura interna del software.

**Figura 4. Pruebas caja negra**



Fuente J.M. Drake (2009)

Las pruebas funcionales o de caja negra son una estrategia para seleccionar las pruebas de fallos basándose en las especificaciones de los componentes y programas, y no del conocimiento de su implementación. El sistema se considera como una caja negra cuyo comportamiento solo se puede determinar estudiando las entradas y de contrastarlas con las respuestas que proporciona el sistema.

Este enfoque se puede aplicar de igual forma a los sistemas organizados como librerías de funciones, o como objetos. El probador introduce las entradas en los componentes del sistema y examina las salidas correspondientes. Si las salidas no son las previstas, entonces la prueba ha detectado exitosamente un fallo en el software.

El problema clave para el probador de defectos es seleccionar la entrada que tienen una alta probabilidad de ser miembro del conjunto Ee. En muchos casos la selección se basa en la experiencia previa de los ingenieros de pruebas. Ellos utilizan el conocimiento del dominio para identificar los casos de prueba que probablemente van a mostrar fallos.

Estas pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa.

Los casos de prueba pretenden demostrar que:

- Las funciones del software son operativas
- La entrada se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa.

Estas pruebas intentan encontrar errores de las siguientes categorías:

- Errores de interfaz
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

## **2.8 MANTENIMIENTO DE SOFTWARE**

Es el proceso de mejora y optimización del software después de su entrega al usuario final es decir con la revisión del programa así, como también corrección de defectos y prevención de los mismos.

### **2.8.1 Tipos de mantenimiento de software**

**Mantenimiento perfectivo:** conjunto de actividades para mejorar o añadir nuevas funcionalidades requeridas por el usuario.

**Mantenimiento adaptativo:** conjunto de actividades para adaptar el sistema a los cambios en su entorno tecnológico.

**Mantenimiento correctivo:** conjunto de actividades dedicadas a corregir defectos en el hardware o en el software detectados por los usuarios durante la exploración del sistema.

**Mantenimiento preventivo:** conjuntos de actividades para facilitar el mantenimiento futuro del sistema.

## **2.9 CALIDAD DE SOFTWARE**

Con el fin de entregar a los clientes productos satisfactorios, el software debe alcanzar ciertos niveles de calidad. Para alcanzar buenos niveles de calidad el número de defectos debe ser mínimo.

Es difícil tener una definición exacta de lo que es la calidad pero existen seis atributos que permiten medirla de una manera sencilla, estos seis atributos son:

**Funcionalidad:** Habilidad del software para realizar el trabajo deseado.

**Fiabilidad:** Habilidad del software para mantenerse operativo.

**Eficiencia:** Habilidad del software para responder a una petición de usuario con la velocidad apropiada.

**Usabilidad:** Habilidad del software para satisfacer al usuario.

**Mantenibilidad:** Habilidad del software para poder realizar cambios en el rápidamente y con una adecuada proporción cambio/costo.

**Portabilidad:** Habilidad del software para correr en diferentes entornos informáticos.

A la hora de abordar estos atributos es importante distinguir entre cuatro conceptos muy relacionados con la mejora de la calidad de software:

**Error:** acción humana que produce una falta.

**Falta:** algo que está mal en un producto (modelo, código, documento, etc.).

**Fallo:** manifestación de una falta.

**Defecto:** Error, falta o fallo

### 3. METODOLOGIA

Para llevar a cabo de manera satisfactoria la actividad de reingeniería era necesario escoger una metodología que se amoldara a las necesidades del software. El modelo cíclico según Pressman, es una cadena de ciclos auto sostenibles que se desarrollan de manera secuencial y lineal, pero que en algunas ocasiones, para un ciclo en particular, el proceso puede terminar después de cualquiera de las actividades que lo componen.

Para el desarrollo de este proyecto se acogió el modelo cíclico, el cual consta de seis actividades. Este capítulo está enfocado al seguimiento de las actividades desarrolladas durante el proceso de reingeniería.

#### 3.1 ANÁLISIS DE INVENTARIOS

Pressman menciona la importancia de esta etapa, ya que a través de su desarrollo se conoce una descripción detallada de las aplicaciones y se definen las que necesitan de reingeniería.

Para la plataforma *Uisport 1.0*, el análisis estuvo enfocado en toda la información básica de la aplicación así:

- Edad
- Población a la que va dirigida
- Funcionalidad
- Objetivo de la creación de la plataforma
- Influencia de la aplicación en los usuarios

#### 3.2 REESTRUCTURACIÓN DE DOCUMENTOS

Pressman ofrece en su libro las soluciones para tres tipos de problemas que se pueden presentar en esta etapa.

Para el caso de la plataforma *Uisport* se contaba con una documentación muy completa que incluía una amplia referencia teórica en la que se justificaba la realización del proyecto, la descripción del portafolio digital, las etapas recorridas en su creación, y la experiencia obtenida durante y después del desarrollo del software.

Mientras se desarrolló la reingeniería de esta herramienta didáctica, se procedió a documentar solo aquellas partes del sistema que experimentaron cambios, y conforme se fueron dando, teniendo en cuenta las recomendaciones dadas por Pressman. (Ver Anexos).

La información obtenida del desarrollo del software en su primera versión, fue de gran ayuda, ya que la mayoría de los sistemas heredados poseen escasa documentación de su desarrollo y experimentación, y consumen más tiempo y más recursos.

### 3.3 INGENIERÍA INVERSA

Las pruebas realizadas para el desarrollo de esta etapa, están descritas en este capítulo en la sección 3.4 *Software pruebas de fase inicial*.

Durante esta etapa se intenta descubrir el objetivo del software: que hace y como lo hace. Mediante la documentación adquirida de la versión 1.0 se pudo obtener cierto tipo de información clave para definir su funcionalidad, pero sin embargo no fue suficiente.

Algunos de los problemas que surgieron en este proceso fueron:

- Falta de comentarios en el código fuente.
- Escasa información de la secuencia de la herramienta.
- Gran cantidad de menús y submenús y poca información de lo que había que hacer en ellos.

Para un análisis profundo se tuvieron en cuenta diferentes tipos de ingeniería inversa, de los cuales se posee todo tipo de documentación:

#### 3.3.1 Ingeniería inversa para comprender datos

Esta etapa de la reingeniería, se realizó a través de las pruebas software escogidas. Para esta sección específica se realizaron pruebas de caja blanca, las cuales son descritas en este capítulo en la sección 3.4.1. *Pruebas de Funcionalidad*.

**Estructura de datos interna:** La ingeniería inversa en esta etapa se enfocó en la revisión exhaustiva del código fuente para poder encontrar y comprender la definición de clases de objetos. Este proceso fue largo y lento debido a la falta de comentarios en el código. Los cambios y mejoras implementados lograron transformaciones en los casos de uso, los cuales se encuentran en anexo 2.

**Estructura de la base de datos:** El análisis realizado a la base de datos fue necesario para comprender las relaciones entre las tablas que contiene y los datos que se ingresan en la plataforma.

Durante este proceso, se evidencio la falta de un ítem en la pestaña *Obtención de Requerimientos*, el cual se creó con el nombre *Enunciado del problema*. Posterior a esto fue necesario adicionar una columna en la tabla *mdl\_block\_Uisportrad* de la base de datos con el nombre *enunciate*, para poder guardar la información ingresada en dicho ítem. (Ver anexo 1)

#### 3.3.2 Ingeniería inversa para entender el procesamiento

Durante esta etapa fue necesario comprender a fondo la funcionalidad del software y sus objetivos, después se buscaron los conflictos existentes entre los aplicativos del sistema y se documentaron las sugerencias y posibles soluciones a estos conflictos.

### 3.3.3 Ingeniería inversa de interfaces de usuario

Para esta etapa fue necesaria la instalación de moodle y de un servidor local. Después se agregó la carpeta Uisport a blocks dentro de la carpeta moodle. Se crearon los dos tipos de usuarios que la herramienta desarrollo: Docente y estudiantes, y se verificaron las funcionalidades de cada una de las pestañas existentes en la herramienta, para cada usuario en particular. En esta etapa de la reingeniería se realizaron pruebas de caja negra con el fin de determinar las mejoras que se debían realizar. Las pruebas implementadas están documentadas en este capítulo, sección 3.4.1 *Pruebas de funcionalidad*.

Las fallas encontradas en la aplicación de dichas pruebas se encuentran documentadas junto a los cambios hechos, en la sección 3.5 *Mejoras realizadas*, de este capítulo.

### 3.4 REESTRUCTURACIÓN

Durante esta fase de la reingeniería se modificó tanto el código como algunos datos del sistema que necesitaban ser actualizados o mejorados. (Ver anexo 4)

- **Reestructuración de código:** Con respecto al código, el análisis fue extenso y bastante tedioso. A pesar de que las clases estaban bien definidas, el hecho de que el código no estuviera comentado, tomó mucho tiempo, así que para empezar a modificarlo fue necesario estar actualizando la carpeta *Uisport* y comparar sus cambios con la base de datos y la plataforma.
- **Reestructuración de datos:** En esta etapa se realizó el análisis de código fuente, el cual se valoró y del que se extrajo información sobre las estructuras de datos definidas en la herramienta.

### 3.5 INGENIERÍA HACIA ADELANTE

La herramienta tecnológica fue puesta a prueba con el fin de comprender el diseño bajo el cual estaba construido. Seguido a esto el software se dividió en partes y se modificó en caso de ser necesario.

Durante este proceso se logró una mayor comprensión del sistema en su parte interna y sus funcionalidades independientes, con lo cual se obtuvo, en algunos aspectos, modificaciones más profundas.

Las mejoras que se sugirieron, fueron realizadas con el fin de complementar la herramienta, y también fueron posibles solo con agregar unas líneas de código.

Durante este y todos los procesos del Modelo Cíclico, se contó con una documentación detallada, la cual se complementó con la obtenida en las pruebas realizadas durante la etapa Ingeniería inversa descrita en este capítulo. La documentación de esta etapa se

encuentra en *Mejoras realizadas* mencionada en la sección 3.5 de este capítulo, y se dividió así:

- Problemas encontrados y errores frecuentes en la funcionalidad de cada pestaña y cada submenú.
- Imagen que soporta cada uno de los errores encontrados.
- Mejoras propuestas no para corregir errores, si no para refinar la calidad del software.
- Sugerencias de las posibles soluciones a los problemas y errores encontrados.
- Respuesta o solución al problema mencionado en la primera parte de esta documentación.

### **3.6 SOFTWARE PRUEBAS DE FASE INICIAL**

Las pruebas de software de la aplicación *Uisport* consistieron en pruebas de funcionalidad para determinar y encontrar errores o falencias en el sitio, que ya se habían evidenciado en el semestre que se utilizó la plataforma. Esta fase del proyecto se dedicó al análisis, e implementación de pruebas de software que permitieron la detección de fallas en las diferentes funciones que se implementaron en el desarrollo de la plataforma verificando así cual es el nivel de calidad en el que se encuentra la herramienta. En primer lugar se hace un análisis de los casos de uso y en base a estos se elaboraron casos de prueba para verificar la funcionalidad de *Uisport*.

#### **3.6.1 Pruebas de funcionalidad**

##### **Pruebas de caja blanca**

Estas pruebas se realizaron con base a la estructura interna del sistema. En primer lugar se revisó el contenido de las páginas y archivos de las diferentes carpetas.

En esta primera revisión se concluyó que *Uisport* es una plataforma que implicó la escritura de suficiente código y un amplio conocimiento del lenguaje por parte de las personas que lo desarrollaron.

El código fue analizado y revisado carpeta por carpeta entendiendo de esta forma las llamadas a funciones, su ubicación dentro del código, las imágenes y los archivos.

##### **Resultados obtenidos:**

- Código poco legible y no documentado.
- Se encontró código que no cumple ninguna función dentro del sistema.
- En el desarrollo en HTML faltan etiquetas lo que no hace funcional la plataforma en algunos navegadores.

- Existen funciones de PHP cuyo uso no es correcto ya que no reciben los parámetros adecuadamente lo que impide el envío y validación de algunos datos.
- Como no existe documentación del código no se puede encontrar referencia a las estructuras de las carpetas, páginas e información que ayude a la comprensión de la estructura del código fuente y que permita realizar un mantenimiento adecuado.
- mala implementación de algunas funciones lo que arrojaba resultados que no eran los esperados. problemas con la codificación de algunos archivos lo que provocaba una visualización incorrecta en algunos navegadores.
- problemas con la redirección a algunas paginas debido a que no se proporcionaban los parámetros necesarios.

### **Pruebas de caja negra**

Estas pruebas consisten en demostrar que cada función del sistema es operativa, debido a que el software se evalúa mediante las respuestas que se puedan obtener a través de una entrada determinada sin necesidad de conocer cómo funciona el código.

Metodología implementada para las pruebas de caja negra:

- Hacer una revisión de las funcionalidades de la aplicación *Uisport*.
- Obtener como resultado de la revisión los casos más relevantes
- Desarrollar y documentar los casos de pruebas según las funcionalidades encontradas.
- Realizar las pruebas dependiendo de cada entrada.
- Revisar las salidas obtenidas y confirmar si cumplen o no con la funcionalidad esperada.

De las funcionalidades generales de la aplicación *Uisport* se escogieron las más relevantes como lo son: Añadir usuarios, verificación, roles, comprobación del bloque portafolio y del funcionamiento de cada una de sus pestañas.

Los casos de prueba se mostraran documentados en la siguiente tabla, especificando funcionalidades del dispositivo *Uisport*, el nombre del caso de prueba según el parámetro a comprobar, los valores de entrada, las respuestas obtenidas y su respectivo resultado.

**Tabla 1. Pruebas de funcionalidad**

Caso de prueba	Funcionalidad	Descripción	Valores de entrada	Respuesta	Resultado
<p>Llenar información solicitada para el desarrollo del proyecto.</p>	<p>Esta opción se encuentra disponible en las pestañas:  Obtención de requerimientos y Diseño.</p>	<p>La pestaña obtención de requerimientos: existen algunas categorías dentro del menú que permiten ingresar toda la información acerca de la primera etapa del proyecto a realizar.  La pestaña de diseño: permite describir la estructura interna del sistema, para ello se debe llenar cada categoría con información.</p>	<p>Se ingresan los datos requeridos en cada pestaña del menú (información acerca de la aplicación que desarrollan los estudiantes durante la asignatura Ingeniería del software I).</p>	<p>Toda la información es guardada exitosamente.</p>	<p><input checked="" type="checkbox"/></p>

Caso de prueba	Funcionalidad	Descripción	Valores de entrada	Respuesta	Resultado
Comprobar el re direccionamiento a otra pestaña del submenú modelos(diagramas/modelos)	Esta opción se encuentra disponible en las pestañas:  Obtención de requerimientos , análisis de requerimientos y diseño.	En el último submenú MODELOS, la opción diagramas/modelos permite ingresar los diagramas necesarios para el software que se está desarrollando.	Se da clic en la opción diagramas/modelos.	La opción diagramas/modelos re direcciona a la pestaña Evidencias, que es la encargada de subir todo tipo de archivos que el estudiante necesite para el desarrollo de su proyecto de clase.	<input checked="" type="checkbox"/>
Generar PDF	Esta opción se encuentra disponible en las pestañas:  Obtención de requerimientos , análisis de requerimientos y diseño.	Existe la opción de generar un reporte consolidado a la fecha en formato PDF con los campos diligenciados por el usuario.	Se da clic en la opción generar.	Se genera el PDF pero no reconoce algunos caracteres.  La información aparece de manera desordenada y poco entendible.  Genera este PDF en la misma ventana que se está trabajando quitando la posibilidad de seguir ingresando o modificando información.	<input checked="" type="checkbox"/>

Caso de prueba	Funcionalidad	Descripción	Valores de entrada	Respuesta	Resultado
Comprobar que la aplicación funcione en los diferentes navegadores	Navegadores más utilizados:  Internet Explorer, Mozilla Firefox y google Chrome.	Utilizar la aplicación en los tres diferentes navegadores, comprobando el funcionamiento de las pestañas.	Ingresar en cada navegador la dirección correspondiente a la aplicación.	La aplicación <i>Uisport</i> trabaja adecuadamente en los navegadores internet Explorer y google Chrome pero en Mozilla Firefox presenta errores.	☒

Caso de prueba	Funcionalidad	Descripción	Valores de entrada	Respuesta	Resultado
Ingreso exitoso	Guardar datos, subir archivos y compartir información.	Estas opciones están disponibles en todas las pestañas.	Ingreso de información, datos y archivos (imágenes, diagramas etc.)	Como respuesta a las acciones anteriormente mencionadas aparece un aviso "ingreso exitoso" pero siempre re direcciona a la pestaña diseño por lo que es difícil saber en qué pestaña estaba trabajando y que ítem es el siguiente a realizar.	☒
Evidencias	Permite subir a la plataforma evidencias.	Estas evidencias pueden ser documentos, diagramas y archivos relacionados con los productos que se generan en cada etapa.	Subir cualquier tipo de evidencia.	Los archivos son subidos exitosamente, se pueden visualizar y comentar.	☑

<b>Caso de prueba</b>	<b>Funcionalidad</b>	<b>Descripción</b>	<b>Valores de entrada</b>	<b>Respuesta</b>	<b>Resultado</b>
Vistas	Esta pestaña permite compartir evidencias.	En una vista se pueden compartir varias evidencias y armar escenarios.	Agregar vista, información sobre la vista y seleccionar la o las evidencias; definir el modo de compartir: acceso interno o externo.	Vista agregada con éxito. Con descripción nombre, fecha y opciones de acceso( con quien esta compartida la evidencia)	<input checked="" type="checkbox"/>
Importar/exp ortar	Esta pestaña permite exportar o importar a formato SCORM las evidencias.	Es posible hacer estas dos acciones con las evidencias que el estudiante desee de tal manera se pueden obtener copias de seguridad del portafolio personal.	Clic en exportar SCROM  Clic importar	Exportar: Se crea el archivo y se tiene la opción de descargarlo.  Importar: Es posible subir un archivo.	<input checked="" type="checkbox"/>
Portafolios compartidos	Esta pestaña cumple con la función de mostrar una lista de las evidencias que han sido compartidas.	Es posible revisar las vistas disponibles hacer comentarios los cuales quedan enlazados en las vistas.	Revisar y escribir comentarios.	Vistas disponibles revisadas y comentadas.	<input checked="" type="checkbox"/>

### 3.7 MEJORAS REALIZADAS

1. Falla: La interfaz era un poco desordenada, el usuario no contaba con ayuda durante la navegación y se perdía fácilmente.

**Figura 5.** Pestaña Obtención de requerimientos versión 1.0

The screenshot displays the 'UISport e-portfolio' interface. At the top, there is a breadcrumb trail: 'Página Principal > Mi Portafolio > Obtención de Req.'. Below this is a navigation menu with options: 'Información', 'Obtención de Req.', 'Análisis de Req.', 'Diseño', 'Evidencias', 'Vistas', 'Exportar/Importar', and 'Portafolios compartidos'. The main heading is 'OBTENCION DE REQUERIMIENTOS'. The text below explains that users should use the 'treeview' menu to fill out the 'Requirements Analysis Document (RAD)'. A list of required items includes: 'Enunciado del problema', 'Descripción de los escenarios', 'Descripción de los casos de uso', and 'Diagramas de casos de uso'. A lightbulb icon is present. Below that, a section titled 'GENERAR REPORTE CONSOLIDADO A LA FECHA' instructs users to save and upload their files, with a 'Generar' button. A toggle switch labeled 'Expandir Colapsar' is shown above a list of items: 'INTRODUCCION', 'SISTEMA ACTUAL', 'SISTEMA PROPUESTO', and 'MODELOS'. A link for 'Ayuda del sistema' is located at the bottom right.

Mejora: Se organizó la interfaz, de manera que el usuario se ubique fácilmente. Se resaltaron las opciones *Ayuda del sistema* y *Generar reporte* y se les agregaron imágenes que llamen la atención.

**Figura 6.** Pestaña obtención de requerimientos versión 2.0



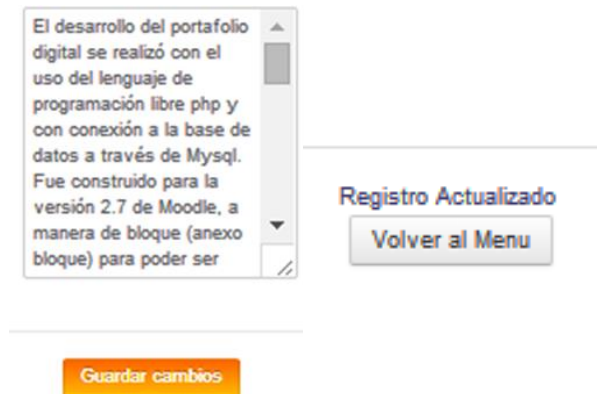
2. Falla: Después de ingresar información en cualquier pestaña, la herramienta envía al usuario a la pestaña Diseño, lo cual lo desubica y no cumple ninguna función.

**Figura 7.** Pestaña Diseño



Mejora: El error fue corregido, eliminando el re direccionamiento directo a la pestaña Diseño. Adicional a esto, después de guardar cualquier tipo de información en cualquier pestaña, el usuario tiene la opción de volver al menú donde se encontraba.

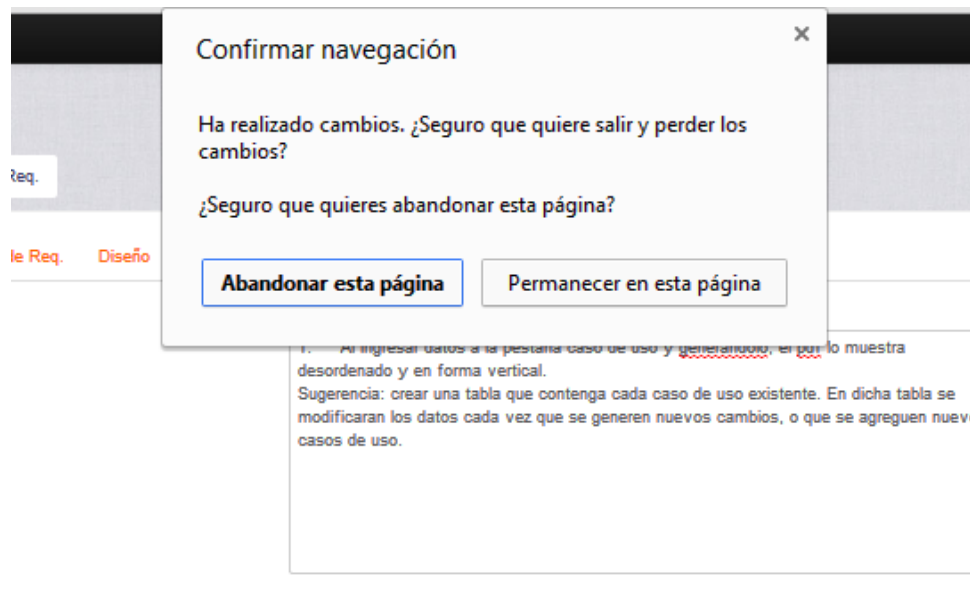
**Figura 8.** Botones Guardar cambios y Volver al menú



3. Falla: Cuando el usuario ingresa información en cualquier pestaña, y se sale sin guardar los cambios, el sistema no tiene ninguna advertencia o sugerencia de que perderá los datos ingresados.

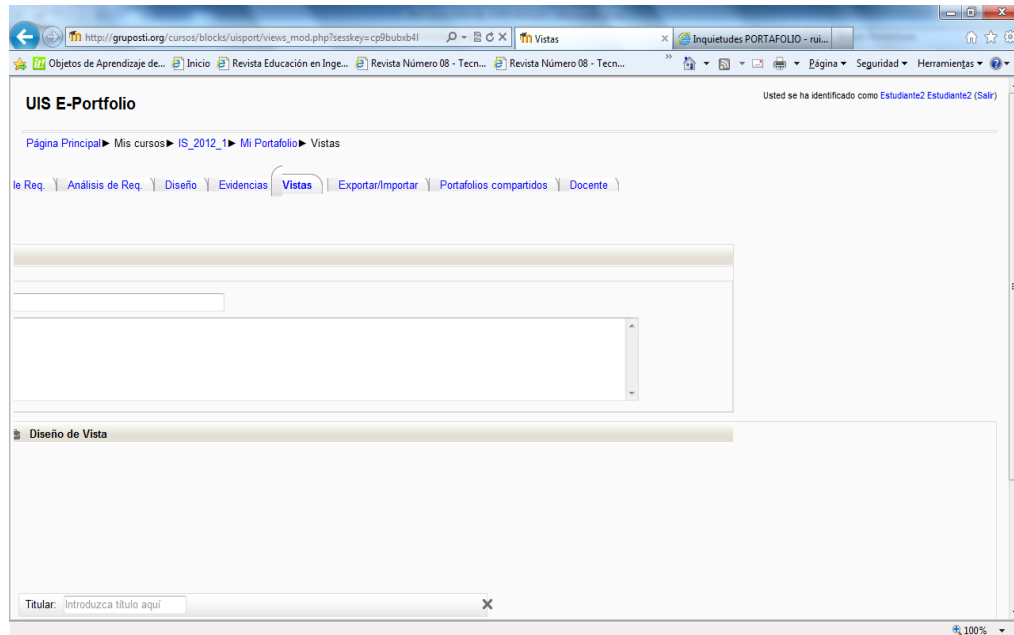
Mejora: El usuario cuenta con dos opciones: Abandonar la página y perder los cambios, o continuar editando y guardar los cambios.

**Figura 9.** Advertencia de abandono de página



4. Falla: Cuando se pasa de una pestaña a otra, y dependiendo del navegador, se desconfiguran los cuadros de texto y el menú desplegable.

**Figura 10. Cuadro de texto desconfigurado**



Mejora: Las pestañas funcionan correctamente, los cuadros de texto salen bien ubicados y el menú desplegable no presenta ningún inconveniente.

**Figura 11. Cuadro de texto nueva versión**

Vista:

Información de la vista

Titulo\*

Descripción

Sus Items

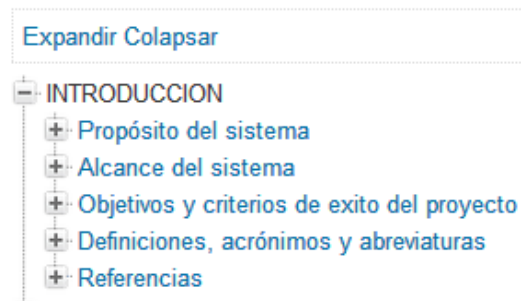
- general
- otro

Diseño de Vista

Titular:  Introduzca título aquí

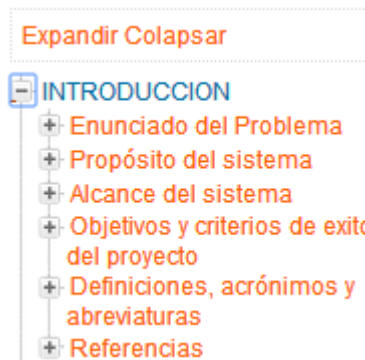
5. Falla: En la descripción de la pestaña Obtención de Requerimientos, se mencionaba que el usuario debía ingresar información acerca del Enunciado del problema, pero en el menú desplegable no se contaba con el espacio para diligenciar este tipo de información.

**Figura 12. Menú desplegable versión 1.0**



Mejora: Se creó en el menú desplegable el ítem Enunciado del Problema.

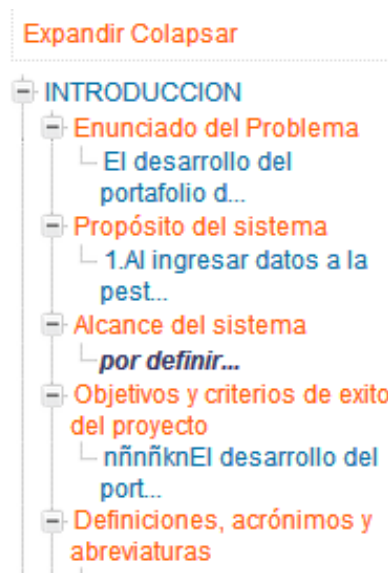
**Figura 13. Enunciado del problema**



6. Falla: En el menú de cualquier pestaña, la opción Expandir solo cumple su función algunas veces, en ocasiones direcciona a otra pestaña o no hace nada.

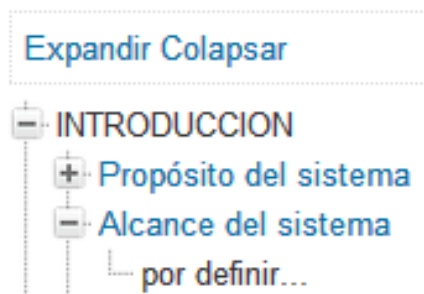
Mejora: Se corrigió el error, y la opción expandir cumple con su función.

**Figura 14. Función expandir**



7. Falla: Cuando en un submenú no se ha ingresado ningún dato, aparece el texto “por definir”, pero no sobresale ni se diferencia del resto de texto.

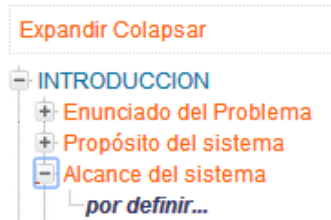
Figura 15. “Por definir” versión 1.0



Mejora: Se dejó el texto “por definir”, pero se puso en negrita y cursiva para que el usuario vea claramente que no ha llenado ese campo.

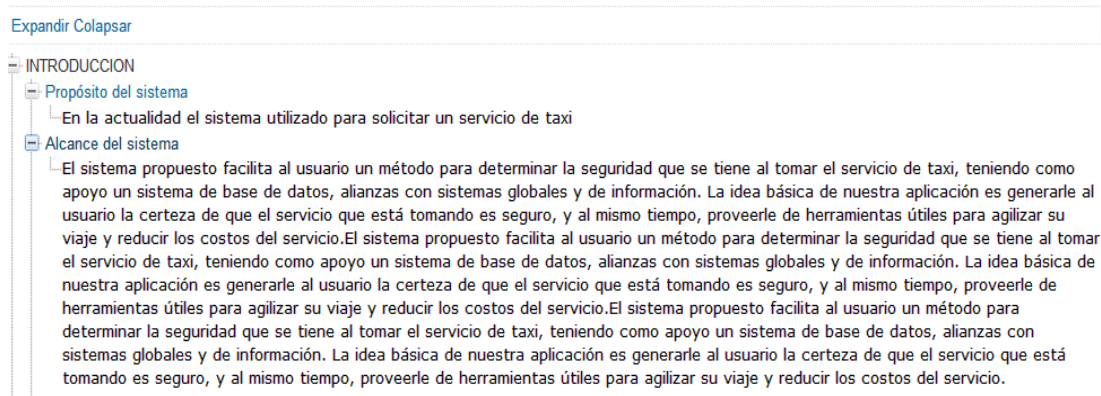
Figura 16. “Por definir” versión 2.0

## OBTENCIÓN DE REQUERIMIENTOS



8. Falla: Después de haber llenado los diferentes campos del menú desplegable, el usuario daba clic en expandir, y el menú le mostraba todo el texto ingresado en cada ítem. Esto era extenso y poco agradable a la vista.

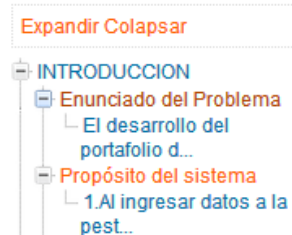
Figura 17. Expandir versión 1.0



Mejora: Cuando el usuario expande el menú, puede ver solo cierta cantidad de texto ingresado seguido de puntos suspensivos, que indican que hay más texto. Si damos clic nos direcciona al cuadro de texto donde se encuentra esta información. Desde allí puede editar o borrar la información. En cuanto a diseño mejoro la plataforma, pues se ve mucho más ordenada.

Figura 18. Expandir versión 2.0

## OBTENCIÓN DE REQUERIMIENTOS



9. En las pestañas *Obtención de Requerimientos* y *Diseño*, el menú está en forma de árbol, y contiene todos los ítems a diligenciar de forma anidada. Cada ítem cuenta con un signo más (+) que permite expandir la información ingresada o en su defecto, el texto “por definir” si no se ha ingresado nada. Un signo menos (-) marca algo ya expandido; con un clic allí se hará colapsar esa rama del árbol.

Falla: El menú de la pestaña *Análisis de requerimientos* no está en forma de árbol, por lo que le resta uniformidad a la plataforma. Se debe contar con el mismo tipo de menú en la plataforma.

Figura 19. Menú Análisis versión 1.0

## ANALISIS DE REQUERIMIENTOS

En esta etapa usted debe realizar el análisis de los requerimientos obtenidos, para ello, debe presentar:

- Modelo de Objetos (Diagrama de clase y sus relaciones)
- Modelos dinámicos (Diagrama de máquina de estado y secuencia)



### GENERAR REPORTE CONSOLIDADO A LA FECHA

No olvide guardar el archivo en su computador y subirlo al sistema utilizando la opción de evidencias. Sea lo mas descriptivo posible en la identificación de las evidencias.

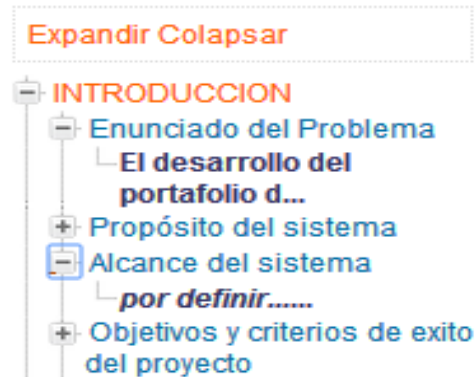
[Generar](#)

INTRODUCCION	SISTEMA ACTUAL	SISTEMA PROPUESTO	MODELOS
<p><b>Propósito del sistema</b></p> <p>En la actualidad el sistema utilizado para solicitar un servicio de taxis el de una llamada telefónica a una empresa de estos, o tomarlo en la calle. El primero tiene una gran desventaja puespor la diversidad de empresas hay una cantidad de números telefónicos que no es fácil memorizar; y la segunda opción, cuando se toma un taxi en la calle, no puede saberse si es un auto legal, con lo cual, estamos expuestos a robos y a arriesgar hasta la propia vida. Este problema de seguridad es de gran controversia hoy en día; a pesar de que se tomen las precauciones necesarias, no se está exentode ser víctimasde asaltos.</p> <p>El sistema propuesto facilita al usuario un método para determinar la seguridad que se tiene al tomar el servicio de taxi, teniendo como apoyo un sistema de base de datos, alianzas con sistemas globales y de información. La idea básica de nuestra aplicación es generarle al usuario la certeza de que el servicio que está tomando es seguro, y al mismo tiempo, proveerle de herramientas útiles para agilizar su viaje y reducir los costos del servicio.</p>			

Mejora: Se implementó el menú de árbol para la pestaña *Análisis de requerimientos*.

Figura 20. Menú Análisis versión 2.0

## ANÁLISIS DE REQUERIMIENTOS

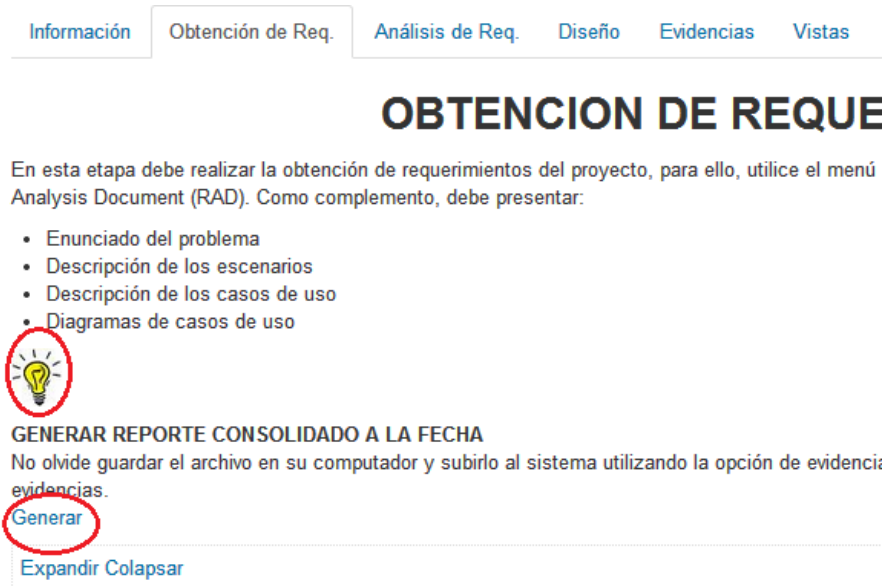


10. Las pestañas *Obtención de requerimientos*, *Análisis de Requerimientos* y *Diseño*; cuentan con dos importantes iconos:

- **Heurístico:** También llamado *Ayuda del sistema*, permite al usuario conocer la etapa que está diligenciando en el desarrollo del software. Proporciona un amplio marco conceptual que ubica al usuario y lo guía en el desarrollo de su proyecto.
- **Generar PDF:** Esta opción permite al usuario visualizar toda la información diligenciada en las diferentes pestañas de la herramienta, creando un documento PDF que recopila y ordena dicha información. Este PDF es el documento final que el usuario Estudiante debe enviar al usuario Docente para que su trabajo sea evaluado.

Falla: Ninguna de las dos opciones es visible para el usuario, pues se confunden con el resto del texto y no poseen la información necesaria para que el usuario las reconozca y las use.

**Figura 21.** Botones Generar y Ayuda del sistema Versión 1.0



Mejora: Se modificó la interfaz para apartar estas dos opciones del resto de la información en la página. Adicional a esto, se asignaron dos imágenes que llaman la atención del usuario y le informa para que sirve cada una.

**Figura 22.** Botones Generar y Ayuda del sistema Versión 2.0



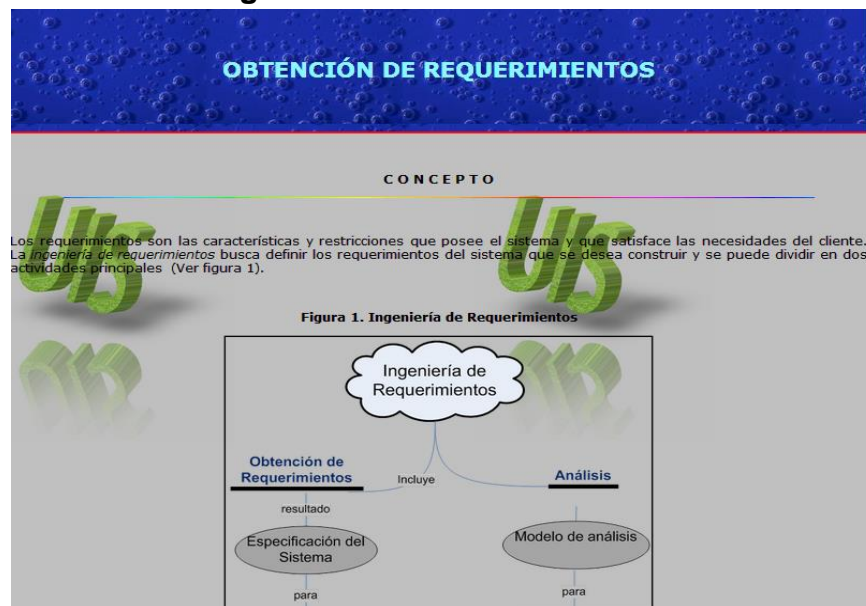
11. Falla: La opción Ayuda del sistema o Heurístico, mencionada anteriormente, se abre en una página web que contiene el marco teórico referente a cada pestaña de la plataforma. En esta página web la información no está centrada, el zoom de la página es alto y no permite visualizar la información completa.

**Figura 23.** Heurístico versión 1.0



Mejora: Se mudó todo el contenido a un nuevo formato de página. La información se ve completa y el zoom es el adecuado.

**Figura 24.** Heurístico versión 2.0



12. Falla: Al generar el pdf, la información insertada en cada ítem del menú desplegable no aparece clara, y se crean caracteres que el usuario no ha ingresado.

**Figura 25.** Caracteres especiales en el pdf versión 1.0

## 1. INTRODUCCION

### 1.1 Propósito del sistema

En la actualidad el sistema utilizado para solicitar un servicio de taxis es el de una llamada telefónica a una empresa de estos, o tomarlo en la calle. El primero tiene una gran desventaja pues por la diversidad de empresas hay una cantidad de números telefónicos que no es fácil memorizar; y la segunda opción, cuando se toma un taxi en la calle, no puede saberse si es un auto legal, con lo cual estamos expuestos a robos y a arriesgar hasta la propia vida. Este problema de seguridad es de gran controversia hoy en día a pesar de que se tomen las precauciones necesarias, no se está exentado de ser víctima de asaltos. El sistema propuesto facilita al usuario un método para determinar la seguridad que se tiene al tomar el servicio de taxi, teniendo como apoyo un sistema de base de datos, alianzas con sistemas globales y de información. La idea básica de nuestra aplicación es generarle al usuario la certeza de que el servicio que está tomando es seguro, y al mismo tiempo, proveerle de herramientas útiles para agilizar su viaje y reducir los costos del servicio.

Mejora: aparecía el carácter &nbsp; que equivale a un espacio en HTML. Se modificó el código y el pdf genera el texto como el usuario lo ingreso.

**Figura 26.** Caracteres especiales en el pdf versión 2.0

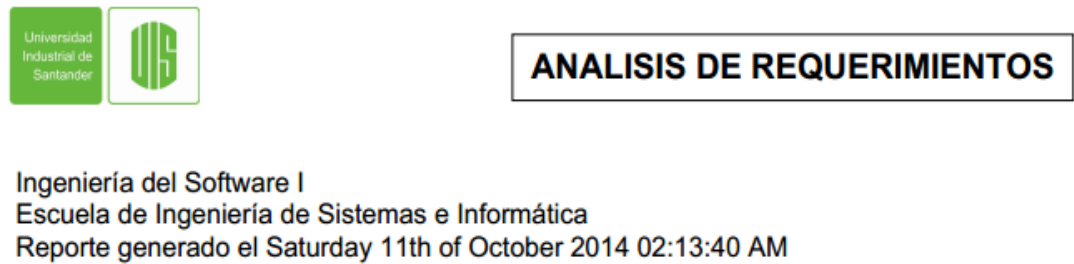
## 1. INTRODUCCION

### 1.1 Enunciado del Problema

El desarrollo del portafolio digital se realizó con el uso del lenguaje de programación libre php y con conexión a la base de datos a través de Mysql. Fue construido para la versión 2.7 de Moodle, a manera de bloque (anexo bloque) para poder ser incorporado dentro de uno o varios cursos. UISPort 2.0 será implementado en el curso Ingeniería del Software I del programa de pregrado Ingeniería de Sistemas e informática de la UIS, dándole continuidad a los objetivos originales de la primera versión. El desarrollo del portafolio digital se realizó con el uso del lenguaje de programación libre php y con conexión a la base de datos a través de Mysql. Fue construido para la versión 2.7 de Moodle, a manera de bloque (anexo bloque) para poder ser incorporado dentro de uno o varios cursos. UISPort 2.0 será implementado en el curso Ingeniería del Software I del programa de pregrado Ingeniería de Sistemas e informática de la UIS, dándole continuidad a los objetivos originales de la primera versión.

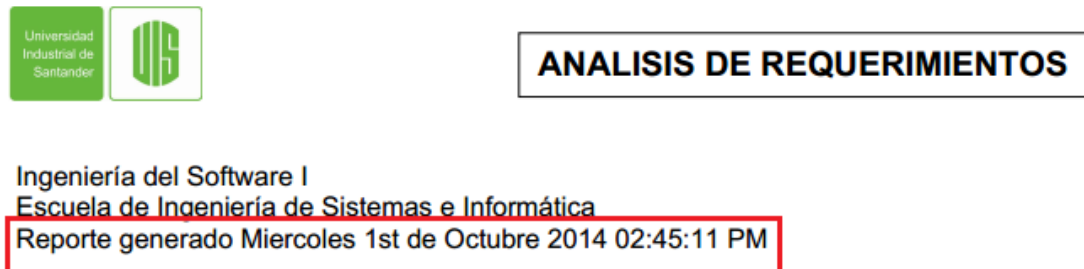
13. Falla: El PDF muestra la fecha y la hora en la que se generó el documento. El problema surge porque la fecha está en inglés, y muchas veces la hora no coincide con la hora real.

**Figura 27.** Fecha y hora en el pdf versión 1.0



Mejora: Los dos problemas quedaron solucionados, y tanto la fecha como la hora coinciden con las actuales.

**Figura 28.** Fecha y hora en el pdf versión 2.0

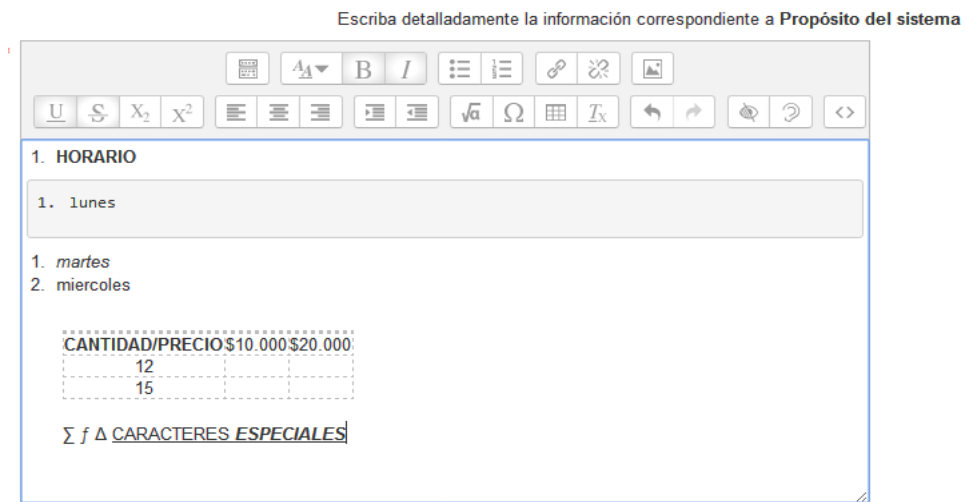


14. Falla: Cuando el usuario genera el PDF, este se abre sobre la misma pestaña en la que esta moodle, por lo que el usuario no puede revisar la herramienta e-portafolio y trabajar en ella. Para ello debe ir atrás y pierde la vista del PDF.

Mejora: se hicieron los cambios respectivos y ahora el PDF se genera en una nueva pestaña. El usuario ahora puede realizar los cambios que crea necesarios en la herramienta e-portafolio, y puede actualizar el PDF para ver reflejados dichos cambios.

15. Falla: La información que aparece en el documento PDF, corresponde a la ingresada por el usuario a través de cuadros de texto que posee cada ítem en el menú desplegable. Estos cuadros de texto le proporcionan al usuario muchas posibilidades tales como: Negrita, cursiva, viñetas, tablas, etc... Todos estos cambios se visualizan en la caja de texto, como se ve en la siguiente imagen:

**Figura 29.** Opciones en el cuadro de texto versión 1.0



Pero en el documento PDF no se ven estos cambios; como se evidencia en la siguiente imagen:

**Figura 30.** Resultado del cuadro de texto en el pdf versión 1.0

**1. INTRODUCCION**

**1.1 Propósito del sistema**

HORARIO

lunes

martes

miercoles

CANTIDAD/PRECIO - \$10.000 - \$20.000

12

15

?&nbsp;?&nbsp;? CARACTERES ESPECIALES

Mejora: Se quitaron todas las opciones en los cuadros de texto, y se puso un sencillo para que el usuario solo ingrese la información, y el documento PDF la ubique por defecto.

**Figura 31.** Opciones en el cuadro de texto versión 2.0


Escriba detalladamente la información correspondiente a Alcance del sistema

Obligatorio

Guardar cambios

16. Falla: Al generar el PDF, la información referente a los casos de uso y a los escenarios aparece de manera vertical, lo cual luce desordenado y con poco espacio entre líneas.

**Figura 32.** Casos de uso versión 1.0



ANALISIS DE REQUERIMIENTOS

Sin escenarios

**5.2 Casos de uso**

caso de uso general

\* *Actores participantes:*  
vendedorcompradorbanco

\* *Flujo de eventos:*  
vendercomprarpagar&nbsp;

\* *Condiciones de entrada:*  
vender producto

\* *Condiciones de salida:*  
pagar producto

**5.3 Diagramas/Modelos**

primero (->Académicos)

se3gundo (->Académicos)

probando (->Académicos)

Mejora: Se creó una tabla para los escenarios, y una para los casos de uso con el fin de que la información se vea más ordenada y fácil de interpretar.

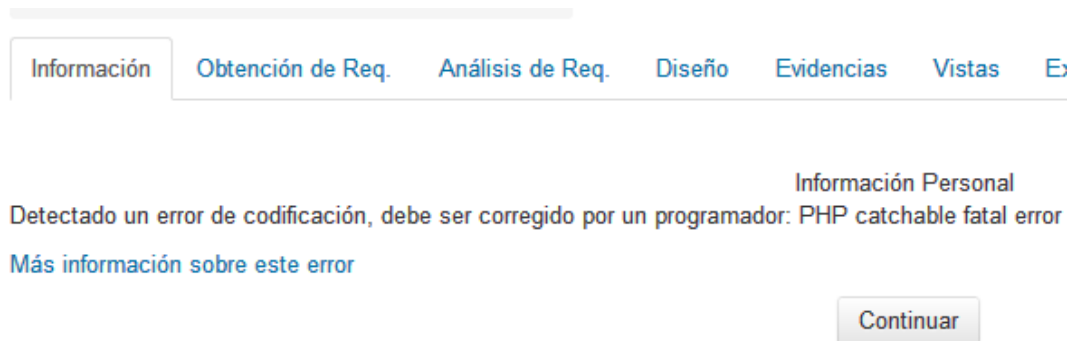
**Figura 33.** Tabla Casos de uso versión 2.0

**5.2 Casos de uso**

Actores participantes	Flujo de eventos	Condiciones de entrada	Condiciones de salida
Usuario	El actor pide el servicio de taxi. La base de datos busca disponibilidad de vehículos. Se retornan los datos de algún vehículo disponible. El usuario confirma la carrera. Se le entregan los datos de ubicación del usuario al taxista.	Tener el acceso con las empresas de taxis. Haber accedido a la opción de pedido de servicio de taxi de la aplicación.	El taxista debe confirmar que le realizó la carrera al usuario que pidió el servicio.
Administrador de la base de datos	El actor se ingresa como usuario privilegiado al sistema de base de datos. El sistema valida los datos y le da ingreso al usuario. El actor ingresa y/o modifica los datos. Se archivan los cambios.	Analizar qué datos que necesitan ingresar a la base de datos.	Presentar una notificación acerca del éxito del guardado de los datos.
uno tres cuatro seis	El desarrollo del portafolio digital se realizó con el uso del lenguaje de programación libre php y con conexión a la base de datos a través de Mysql. Fue construido para la versión 2.7 de Moodle, a manera de bloque (anexo bloque) para poder ser incorporado dentro de uno o varios cursos. UISPort 2.0 será implementado en el curso Ingeniería del Software I del programa de pregrado Ingeniería de Sistemas e informática de la UIS, dándole continuidad a los objetivos originales de la primera versión.	UISPort 2.0 será implementado en el curso Ingeniería del Software I del programa de pregrado Ingeniería de Sistemas e informática de la UIS, dándole continuidad a los objetivos originales de la primera versión.	Presentar una notificación acerca del éxito del guardado de los datos.

17. Falla: en la pestaña Información, después de ingresar los datos y guardarlos, aparecía un error de codificación.

**Figura 34.** Error en pestaña información



Mejora: Se corrigió el error. Se guarda la información correctamente.

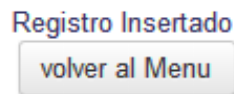
**Figura 35.** Pestaña información corregida



18. Falla: después de ingresar información en cada ítem del menú desplegable, el usuario no cuenta con la opción de volver al menú donde se encontraba.

Mejora: se crearon botones que le permiten al usuario regresar al menú anterior.

**Figura 36.** Botón volver al menú



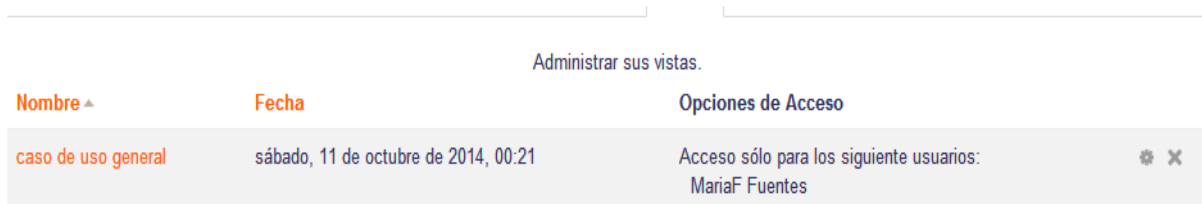
19. Falla: en las pestañas Evidencias y Vistas, el usuario puede ver la lista de evidencias y de vistas que tiene, y al lado derecho tiene la opción de editarlas o borrarlas. El problema surge porque las imágenes asignadas a estas opciones no aparecen. Además las tool tips no estaban funcionando.

**Figura 37.** Iconos y tool tips versión 1.0



Mejora: Debido a la versión de Moodle, se estaban referenciando imágenes que no existían. Se arreglaron los tool tips.

**Figura 38.** Iconos y tool tips versión 2.0



20. Falla: En la pestaña Evidencias, el usuario puede ver una lista de todas las que ha creado, las puede editar y borrar. Si el usuario desea compartir este archivo con otro usuario de moodle debe dirigirse a la pestaña Vistas.

Mejora: con la intención de facilitarle las cosas al usuario, se creó un icono adicional, que le permite desde la lista de todas las evidencias, compartir las que desee. Es decir, si el usuario le da clic en el icono compartir, se direcciona a la pestaña vistas para que comparta el archivo.

**Figura 39.** Icono compartir



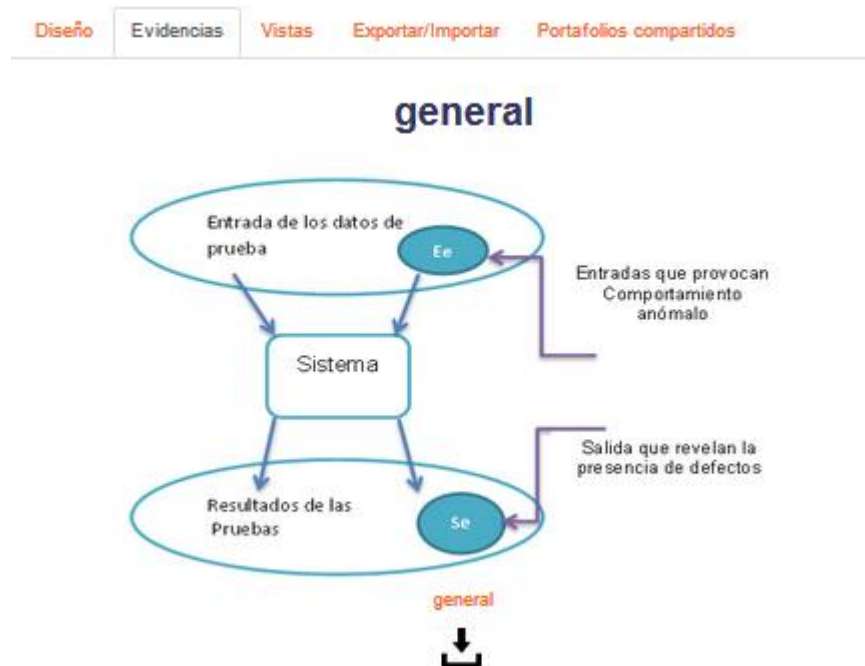
21. Falla: En la pestaña Evidencias, el usuario sube una imagen al sistema y puede compartirla si lo desea. La imagen solo se puede hacer clic sobre el nombre dado, y se abre en otra ventana. No existe la opción de descargar.

**Figura 40.** Opción descargar versión 1.0



Mejora: Se creó la opción de ver la imagen en la misma ventana y sin necesidad de descargarla. También se creó la opción de descargar la imagen, o cualquier archivo que el usuario suba o comparta, si lo desea.

**Figura 41.** Opción descargar versión 2.0



22. Falla: En la pestaña Vistas, si el usuario desea compartir alguna evidencia con los miembros de su curso,

**Figura 42. Roles versión 1.0**

Compartir: Acceso Interno: Acceso sólo para los siguiente usuarios

Acceso Externo

Acceso Interno

Acceso de **todos**

Acceso sólo para los siguiente usuarios

**prueba1**

Nombre	Role
<input type="checkbox"/> MariaF Fuentes	
<input type="checkbox"/> Fabio gutierrez	
<input type="checkbox"/> mariela mantilla	

Agregar

Usted se ha identificado como [Evelin rey \(Salir\)](#)  
[Página Principal](#)

Mejora: Se corrigió el error, y ahora el usuario ve los roles de cada uno de los integrantes de los cursos en los que está matriculado.

**Figura 43. Roles versión 2.0**

Compartir: Acceso Interno: Acceso sólo para los siguiente usuarios

Acceso Externo

Acceso Interno

Acceso de **todos**

Acceso sólo para los siguiente usuarios

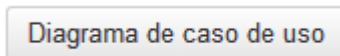
**prueba1**

Nombre	Role
<input type="checkbox"/> MariaF Fuentes	editingteacher
<input type="checkbox"/> Evelin rey	student
<input type="checkbox"/> mariela mantilla	student

23. Falla: Si el usuario desea crear un diagrama de caso de uso, debe dirigirse a la pestaña Evidencias.

Mejora: Después de crear un caso de uso, aparece la opción “Diagrama de caso de uso” la cual direcciona a evidencias, para allí subir el diagrama correspondiente, además se le asigna el título del caso de uso por defecto. Esto facilita la navegación.

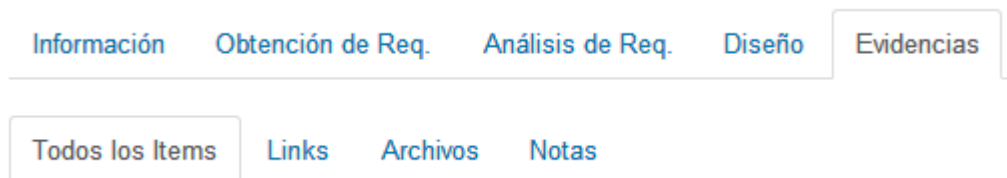
**Figura 44.** Botón Diagrama de caso de uso



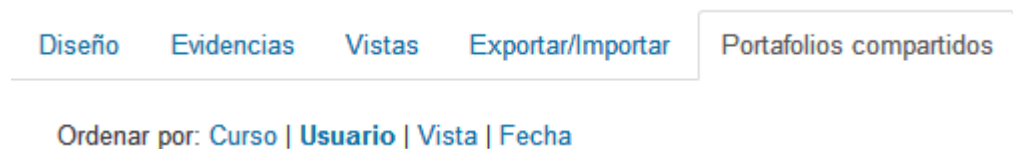
Fuentes autores

24. Falla: Las pestañas Evidencias y Portafolios compartidos no tienen el mismo tipo de menú.

**Figura 45.** Menú Evidencias versión 1.0

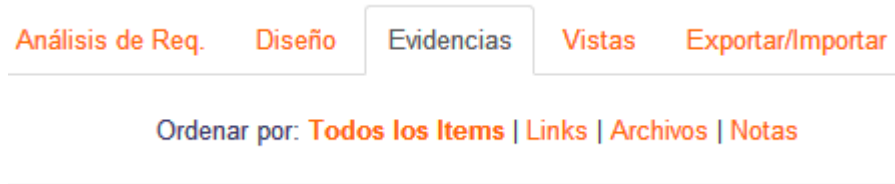


**Figura 46.** Menú portafolios compartidos versión 1.0



Mejora: se implementó el mismo tipo de menú para conservar la uniformidad en la herramienta.

**Figura 47.** Menú Evidencias versión 2.0



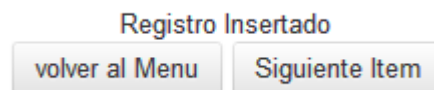
**Figura 48.** Menú Portafolios compartidos versión 2.0



25. Falla: Después de ingresar información y guardar los cambios en cualquier ítem, se creó el botón *Volver al menú*. Pero si el usuario quería seguir editando ítems de ese mismo menú la opción no era eficiente.

Mejora: junto al botón volver al menú se creó la opción siguiente ítem. Su función es que el usuario continúe ingresando información en cada uno de los ítems del menú correspondiente sin tener que ir al menú principal.

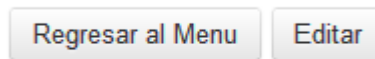
**Figura 49.** Botón siguiente ítem



26. Falla: Desde la pestaña *Análisis de requerimientos* el usuario solo tiene la opción de analizar la información ingresada en la pestaña *obtención de requerimientos* y no la puede editar.

Mejora: se creó un botón en la pestaña *análisis de requerimientos*, que permite editar la información re direccionándonos a la pestaña *obtención de requerimientos*.

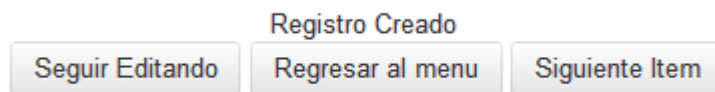
**Figura 50.** Botón editar desde análisis



27. Falla: Después de crear un escenario y guardar los cambios, el usuario debía volver al menú principal para crear un nuevo escenario.

Mejora: Se creó el botón *seguir editando* que permite al usuario crear un nuevo escenario sin volver al menú principal.

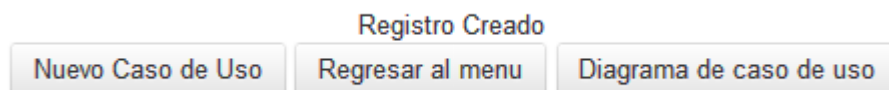
**Figura 51.** Botón seguir editando



28. Falla: Después de crear un caso de uso, el usuario debía volver al menu principal para crear uno nuevo.

Mejora: Se creó el botón *nuevo caso de uso* para que el usuario no tenga que volver al menú principal.

**Figura 52.** Botón nuevo caso de uso



29. Falla: Cuando el usuario daba clic sobre el ítem *objetivos y criterios* desde la pestaña *Análisis de requerimientos*, lo direccionaba a la pestaña *evidencias*, por lo tanto era un error.

Mejora: Se corrigió el error, y ahora si el usuario da clic sobre el ítem *objetivos y criterios*, puede ver la información allí ingresada.

#### 4. DESCRIPCIÓN DE LA NUEVA HERRAMIENTA

La herramienta didáctica *Uisport 2.0* cuenta con una amplia documentación de la versión anterior, por lo cual se tuvo suficiente información para conocer a fondo el software desarrollado, su marco teórico, sus funcionalidades y el público a quien iba dirigido.

Para determinar los procesos de la herramienta que iban a ser modificados o refinados, se hizo un análisis y un estudio mediante el Modelo Cíclico de reingeniería, en el cual se documentó cada actividad realizada como se mencionó en el capítulo 3. *Metodología*.

Durante este proceso se evaluaron y analizaron aspectos determinantes de la primera versión de la herramienta, mediante los cuales se propusieron y ejecutaron cambios trascendentales tanto en el código como en la base de datos.

La descripción de la base de datos de la herramienta *Uisport 2.0* está disponible en el Anexo 1, diagrama de entidad relación.

Cabe resaltar que los diagramas de casos de uso describen cada una de las etapas correspondientes al desarrollo de software, implementadas en la herramienta. Estos diagramas se encuentran en el anexo 2.

Mediante pruebas de caja negra y caja blanca, se realizaron las respectivas sugerencias o soluciones tentativas para refinar los procesos que presentaron problemas, y se contó con la documentación pertinente de este proceso. La descripción de las mejoras realizadas a nivel de diseño e interfaz son enunciadas en el capítulo 3 *Metodología*, sección 3.5 *Mejoras realizadas*.

Dichas mejoras no solo son cambios en el código de la herramienta si no transformaciones completas, por lo tanto fue necesario realizar un manual de usuario para ubicar a quienes experimentaran con la nueva herramienta, *Uisport 2.0*. Este manual está en el Anexo 3 del presente documento.

Con respecto a las líneas de código modificadas y agregadas para realizar los cambios y mejoras correspondientes se construyó un manual técnico que se encuentra en el Anexo 4. Este manual cuenta con la respectiva descripción de los cambios con el fin de ubicar a los futuros desarrolladores de una nueva versión de *Uisport*.

El desarrollo del portafolio digital se realizó con el uso del lenguaje de programación libre php y con conexión a la base de datos a través de Mysql. Fue construido para la versión 2.7 de Moodle, a manera de bloque para poder ser incorporado dentro de uno o varios cursos. Para ubicar al usuario que desee instalar el bloque, se creó un manual que describe una serie de pasos a realizar para la correcta instalación, el cual se encuentra en el anexo 5.

Uisport 2.0 será implementado en el curso Ingeniería del Software I del programa de pregrado Ingeniería de Sistemas e informática de la UIS, dándole continuidad a los objetivos originales de la primera versión.

Con respecto a diferentes aspectos como la seguridad, el acceso a la plataforma y la administración de esta, vienen determinados por Moodle. Los usuarios están regidos a los permisos asignados por los roles de Moodle dentro de un curso.

La plataforma en la versión 1.0 se desarrolló basándose en la herramienta didáctica *portafolio*, aplicándola en las etapas que involucran los procesos de diseño de software; estos aspectos no fueron modificados en la segunda versión.

Uisport 2.0 cuenta con una nueva interfaz, y modificaciones en sus funcionalidades con el objetivo de hacer más fácil y didáctico el paso por la herramienta, para los usuarios.

Las imágenes de la plataforma aparecen a continuación:

- |                                      |                                  |
|--------------------------------------|----------------------------------|
| <b>1</b> Información                 | <b>5</b> Evidencias              |
| <b>2</b> Obtención de requerimientos | <b>6</b> Vistas                  |
| <b>3</b> Análisis de requerimientos  | <b>7</b> Exportar/importar       |
| <b>4</b> Diseño                      | <b>8</b> Portafolios Compartidos |

**Figura 53.** Descripción pantalla principal



**Figura 54.** Ayuda del sistema

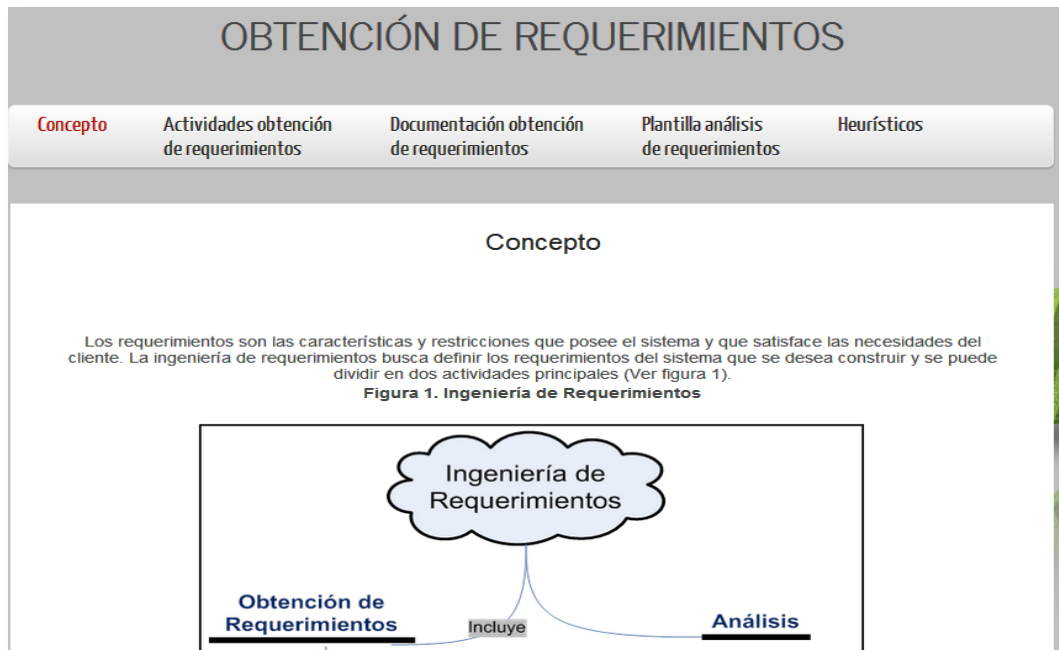


Figura 55. PDF generado

Universidad Industrial de Santander

**ANALISIS DE REQUERIMIENTOS**

Ingeniería del Software I  
Escuela de Ingeniería de Sistemas e Informática  
Reporte generado Domingo 19th de Octubre 2014 12:29:28 PM

Maria Fernanda Fuentes Mantilla

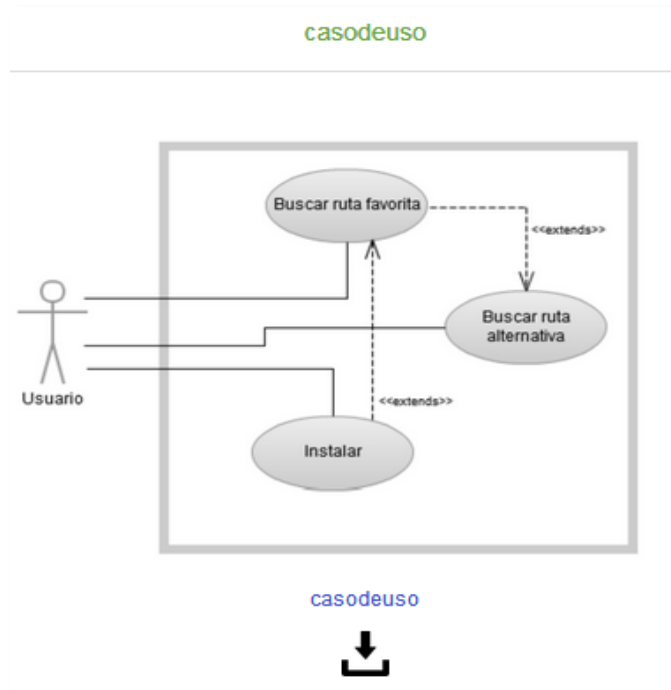
**1. INTRODUCCION**

**1.1 Enunciado del Problema**  
nayibe

**1.2 Propósito del sistema**  
Desarrollar Localicel: una aplicación para el sistema operativo Android mediante la cual el usuario puede descargar para su teléfono móvil la información del tráfico vehicular de la ciudad en la que se encuentra.

**1.3 Alcance del sistema**  
Localicel muestra la ubicación en la que se encuentra el usuario, así mismo el usuario ingresa el lugar de destino, para que Localicel encuentre la ruta más óptima entre los dos puntos, teniendo en cuenta los siguientes factores: menor distancia y menor porcentaje de tráfico según la base de datos de tráfico local.

Figura 56. Subir evidencia



## 5. PRUEBA PILOTO

La metodología implementada para el desarrollo de este proyecto permitió ir evaluando cada aspecto de la herramienta Uisport 2.0 a medida que se le realizaban las mejoras. Después de implementados los cambios tanto en el código como en el diseño de la interfaz, se obtuvo un producto final que cumplía con los requerimientos funcionales iniciales.

No obstante, era indispensable y se estableció como objetivo de este proyecto realizar una prueba piloto para verificar los requerimientos de la nueva versión.

La herramienta *Uisport* fue creada como una herramienta didáctica para el apoyo de la asignatura Ingeniería del software I del programa de pregrado de Ingeniería de sistemas e informática de la Universidad Industrial de Santander, por ello la prueba piloto se llevó a cabo con dieciocho (18) estudiantes de dicho curso correspondiente al segundo semestre del 2014 y se contó con el apoyo de docentes de la asignatura. Al final de la prueba se realizó una encuesta con el fin de determinar las apreciaciones de los estudiantes, el formato de esta se puede observar en el Anexo6 .La herramienta Uisport 2.0 se instaló en el portal del Centro de Desarrollo para la Docencia en la UIS (CEDEDUIS) cuya dirección es <http://tic.uis.edu.co>, que en estos momentos administra y apoya virtualmente los cursos impartidos de diferentes asignaturas en pregrado.

La prueba se realizó con los siguientes pasos:

- Presentación del objetivo de la prueba piloto
- Aporte de la herramienta a la asignatura Ingeniería de software I
- Explicación de la herramienta
- Uso de la herramienta por parte de los estudiantes, tomando como base unos ejercicios guías que les fueron facilitados.
- Sesión de preguntas y observaciones de los usuarios
- Diligenciamiento de la encuesta<sup>12</sup>.( Ver anexo 6)

A continuación se muestra el registro fotográfico de la prueba.

---

<sup>12</sup> Basada en la metodología de GALVIS PANQUEVA, Alvaro. Ingeniería de software educativo- Parte 2. Bogotá: Ediciones Uniandes, 1992.

**Figura 57.** Estudiantes entrando a la plataforma



**Figura 58.** Estudiantes creando un proyecto en la plataforma



**Figura 59.** Estudiantes llenando la encuesta

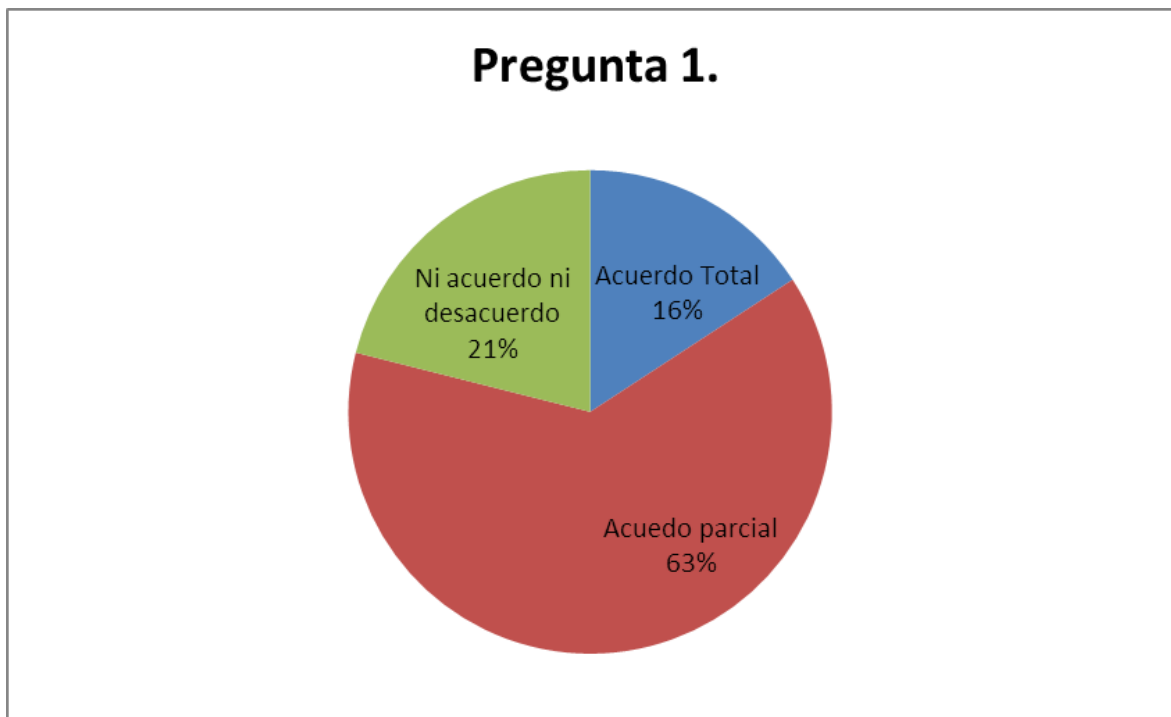


## 5.1 ANALISIS DE LOS RESULTADOS

El análisis se hizo basado en cada una de las preguntas de la encuesta. Los estudiantes respondieron según la experiencia personal que tuvieron con la herramienta. También se tuvo en cuenta cada una de las observaciones realizadas por ellos, durante el desarrollo de la actividad.

- 1) *Los contenidos presentados en la herramienta son suficientes y pertinentes para trabajar el proceso de desarrollo de software.*

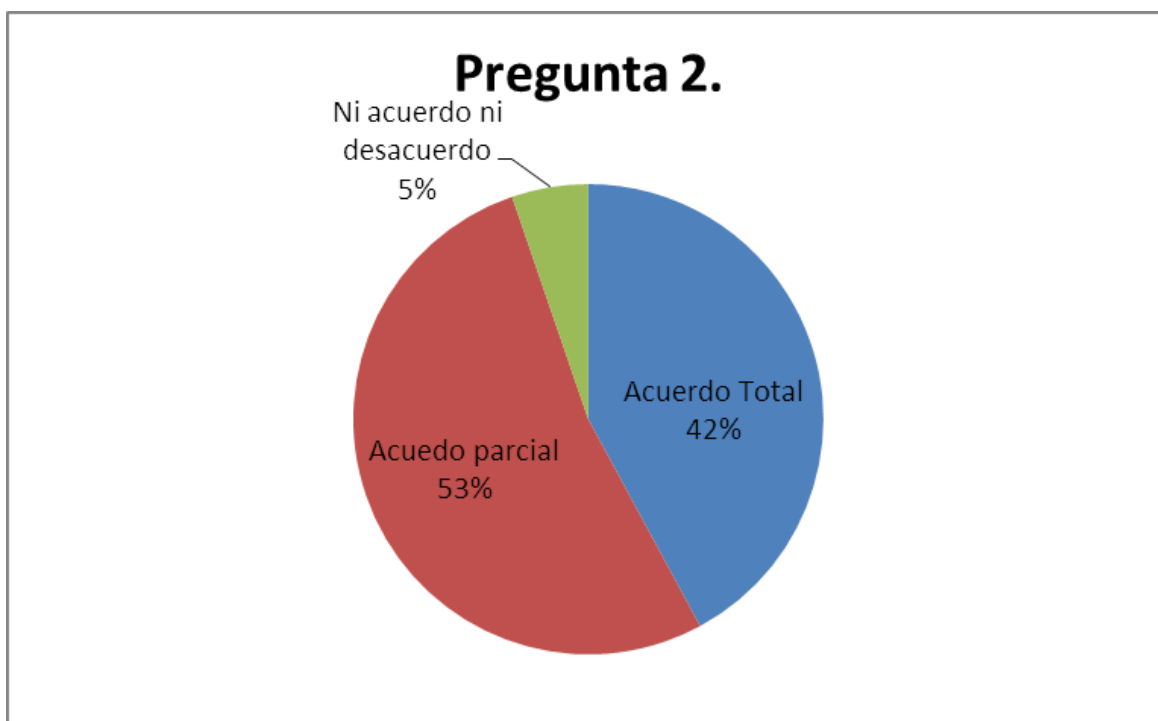
**Figura 60.** Análisis pregunta 1



El resultado con respecto a los contenidos de la herramienta fue positivo. La figura 53 muestra la respuesta de los usuarios, la cual permite afirmar que la herramienta cuenta con la información necesaria para el proceso de desarrollo de software, con miras a un mejoramiento futuro, ya que el 63% de los usuarios está conforme pero el 21% espera un progreso.

2) *Utilizar el programa facilita el aprendizaje de los contenidos de la asignatura de Ingeniería de Software*

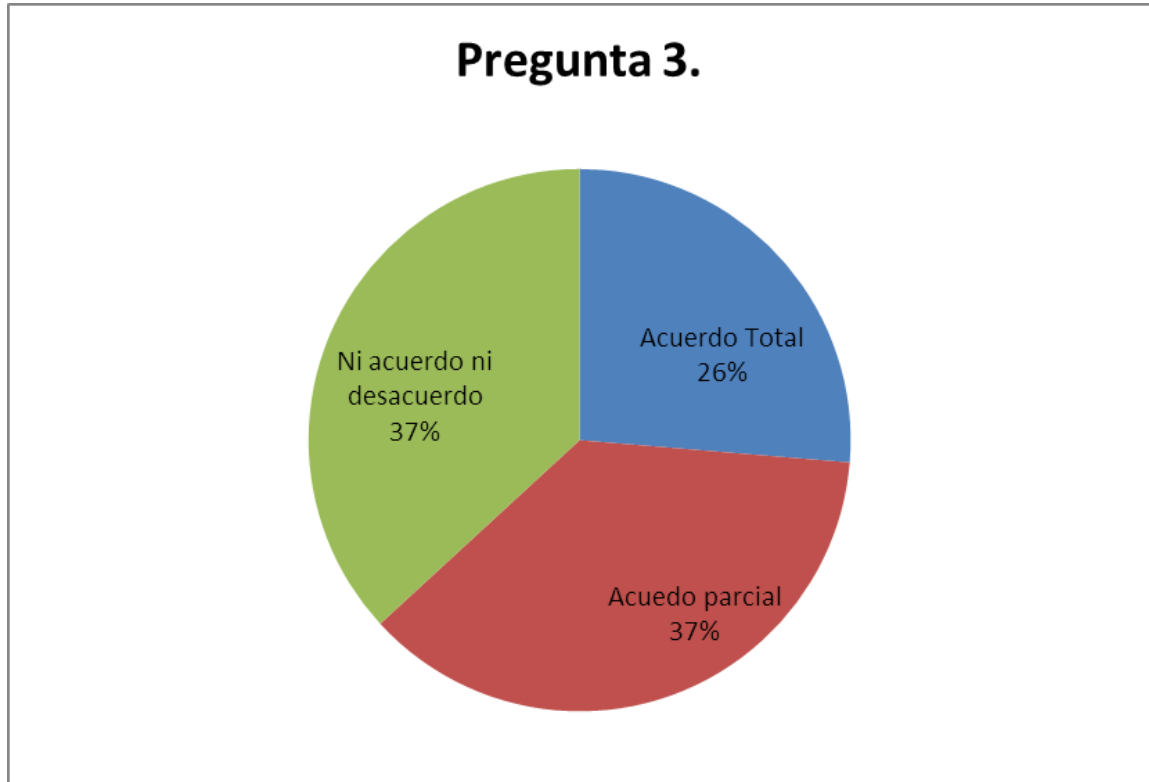
**Figura 61.** Análisis Pregunta 2



Los estudiantes respondieron afirmativamente, reconociendo la herramienta Uisport 2.0 como un facilitador para el aprendizaje de la asignatura Ingeniería de software I. En la figura 61 se evidencio la respuesta positiva de la mayoría de los estudiantes encuestados.

3) *La herramienta permite ser autónomo en el proceso de aprendizaje*

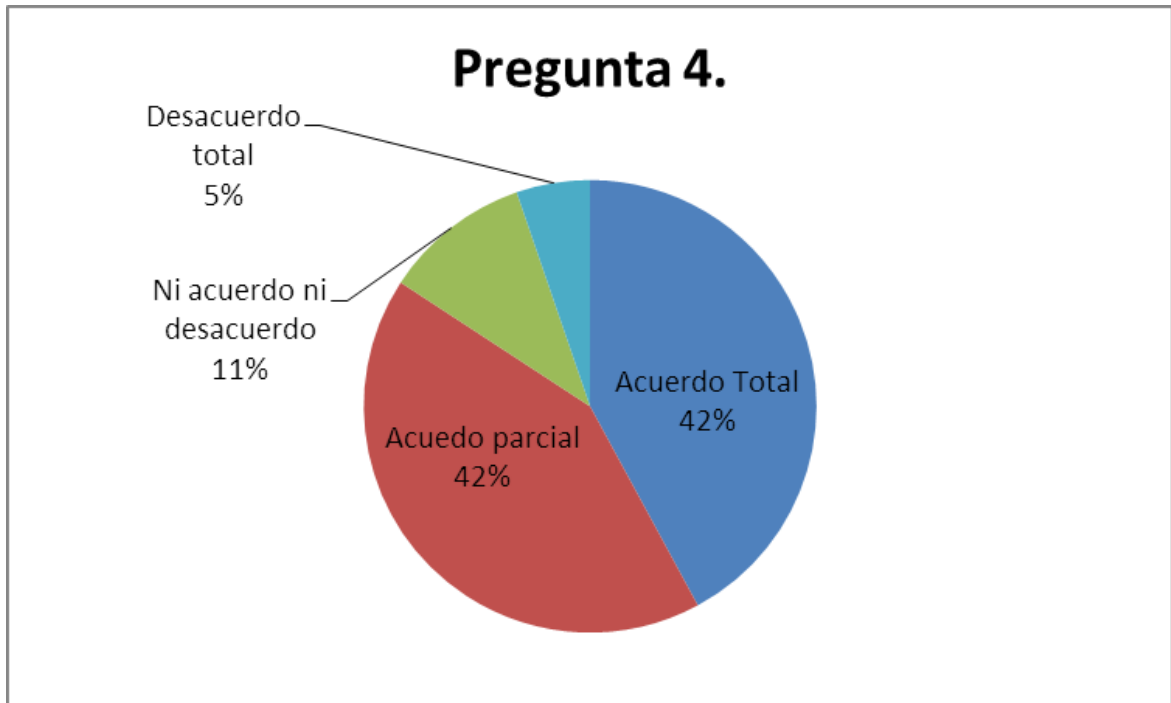
**Figura 62.** Análisis Pregunta 3



Un gran porcentaje de los estudiantes estuvo de acuerdo con la autonomía de la herramienta en el proceso de aprendizaje, pero el 37% no están Ni en acuerdo ni en desacuerdo, lo que permite afirmar que la herramienta aún puede ser mejor en este aspecto. Estos resultados pueden verse en la figura 62.

4) Las evidencias que se pueden registrar en cada etapa del proceso de software son suficientes

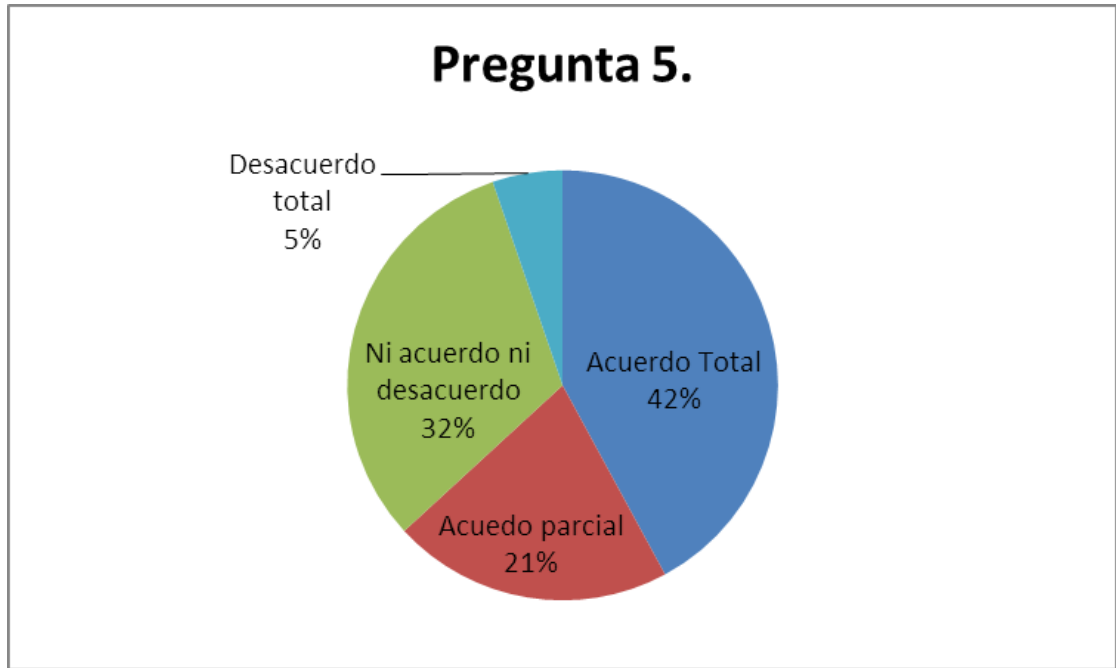
Figura 63. Análisis Pregunta 4



La pregunta 4 ubica al usuario en la pestaña Evidencias, y según los resultados de la encuesta el 42% de los estudiantes están satisfechos con la cantidad de evidencias que el software permite registrar. Otro 42% está conforme, y solo un 11% la acepta pero piensa que debe mejorar. Los resultados expuestos en la figura 63 nos permiten ver también que el 5% de los usuarios encuestados no están satisfechos con esta función de la herramienta, por lo tanto se recomienda un mejoramiento futuro, para próximas versiones de la herramienta.

5) *La herramienta apoya las diferentes etapas del proceso de desarrollo de software*

**Figura 64.** Análisis Pregunta 5

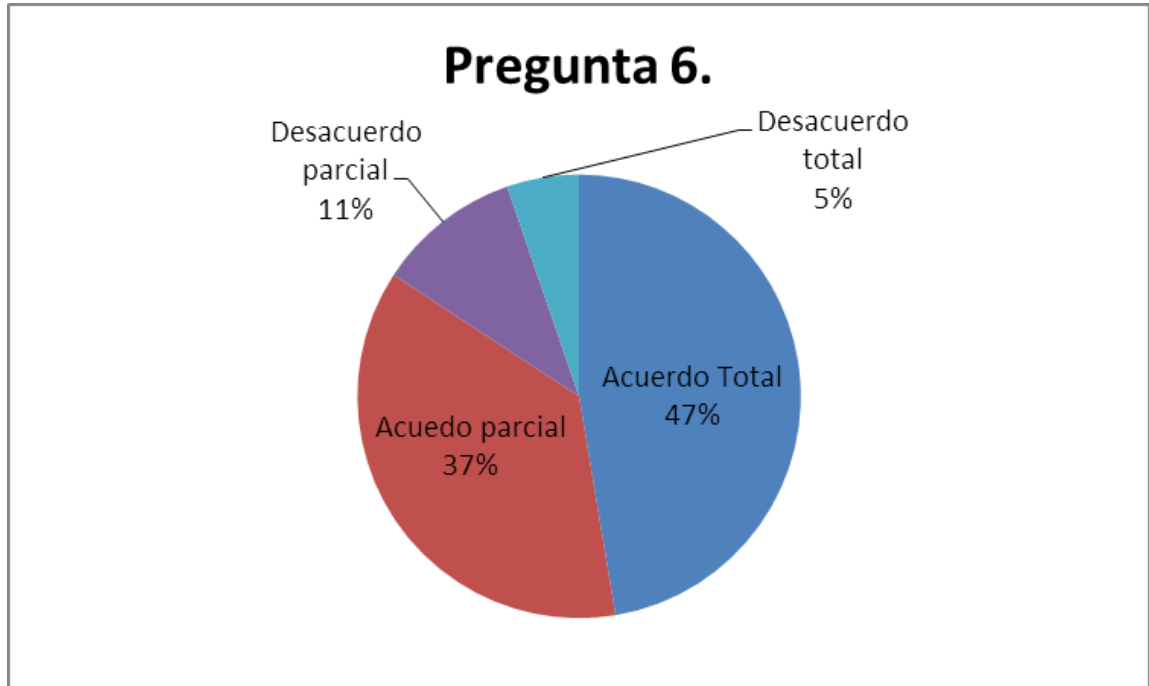


Fuentes autores

Una de las principales funciones de la herramienta pedagógica es la de servir de apoyo a todo el proceso de desarrollo de software, guiando paso a paso al usuario. En general, y como se puede evidenciar en la figura 64, la respuesta de los estudiantes fue positiva, dejando ver un 32% de usuarios que buscan el mejoramiento de la herramienta, pero que están conformes con la versión actual. Solo el 5% de los usuarios encuestados no están satisfechos con el apoyo brindado.

6) *La herramienta permite el trabajo en equipo*

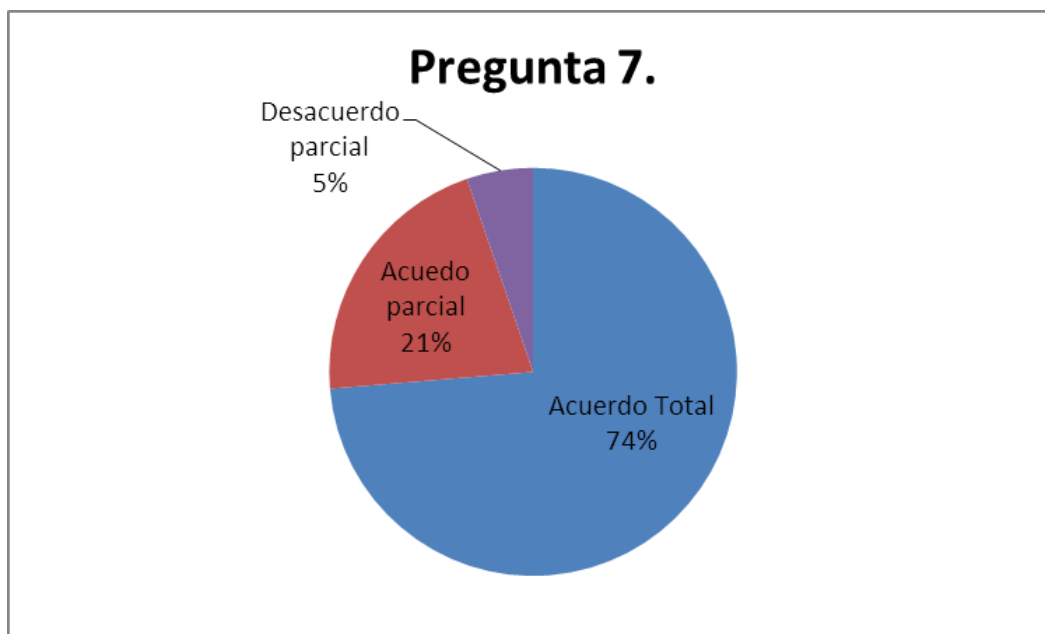
**Figura 65.** Análisis Pregunta 6



La figura 65 nos permite afirmar que los usuarios están satisfechos con los métodos que usa la herramienta para contribuir al trabajo en equipo, pues un 84% de los usuarios encuestados están en Acuerdo total y parcial con esta funcionalidad. Los usuarios no conformes no superan el 16%, pero este porcentaje permite afirmar que se debe buscar un mejoramiento en este aspecto.

7) *La herramienta permite la comunicación con el docente*

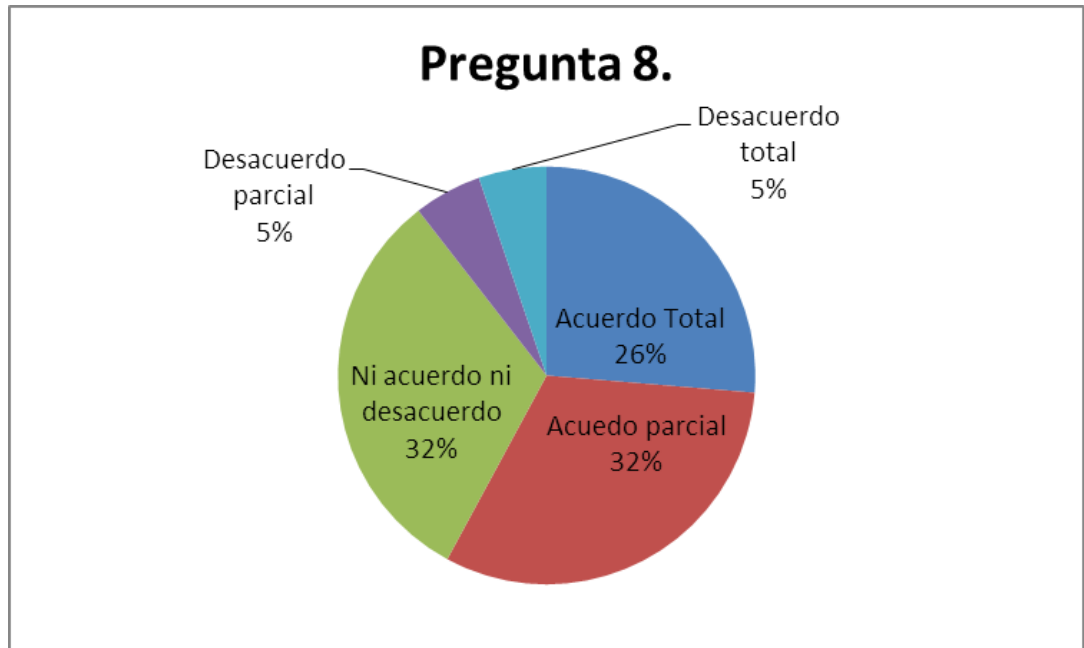
**Figura 66.** Análisis Pregunta 7



Una de las funcionalidades de mayor acogida, fue la que permite la comunicación con el docente. A través de la figura 66 se puede observar que los usuarios encuestados determinaron Acuerdo total con un 74% y Acuerdo parcial con un 21%. Esto permite afirmar que se debe seguir trabajando en la interacción docente-estudiante, pues es un aspecto muy valorado. Teniendo en cuenta las opiniones de todos los usuarios, se hace necesario continuar con el mejoramiento de este aspecto, pues el 5% estuvo en desacuerdo parcial.

8) Las ayudas de la herramienta son consistentes y apoyan el uso de Uisport

Figura 67. Análisis Pregunta 8

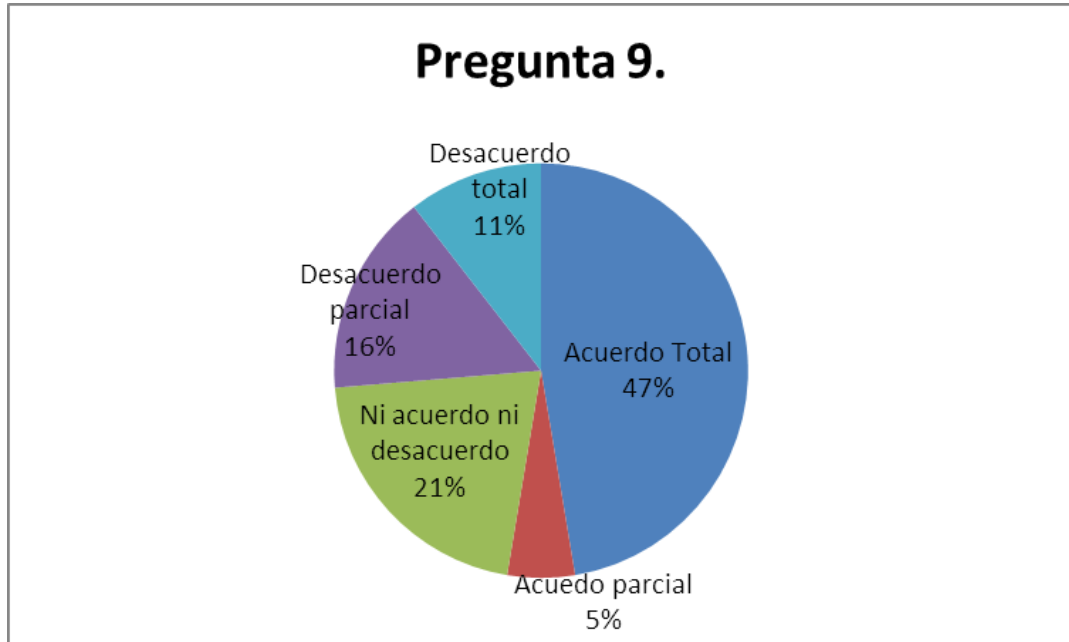


La pregunta 8. Ubica al usuario en la ayuda del sistema, y en que tanto apoyo esta brinda al desarrollo de software a través de Uisport. Los porcentajes en la figura 67 muestran que la cantidad de usuarios totalmente de acuerdo disminuye, y que un 32% están conformes con la herramienta pero recomiendan un mejoramiento en este aspecto.

El porcentaje de usuarios en desacuerdo aumento con respecto a las otras preguntas, lo que permite afirmar que se debe trabajar más a fondo en el desarrollo de las ayudas brindadas.

9) La interfaz es de fácil navegabilidad

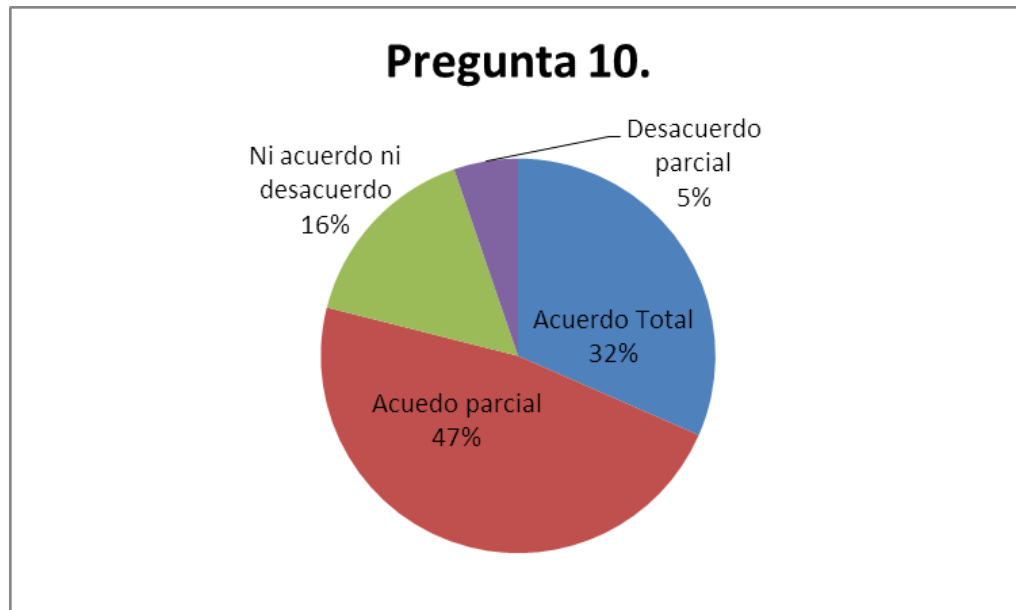
Figura 68. Análisis Pregunta 9



Con respecto a la navegabilidad, la satisfacción de los usuarios vario, como se evidencia en la figura 68. Con un acuerdo total del 47% y un acuerdo parcial de 5%, se puede decir que la mitad de los usuarios encuestados quedo complacido. El 21% de usuarios piensa que está bien pero que puede mejorar. El 16% encuentra poca satisfacción y bastante mejora por realizar y solo un 11% no consideran agradable navegar por la plataforma. En síntesis, la mitad de los usuarios piensa que se le deben realizar grandes mejoras a la plataforma en este aspecto.

**10) La herramienta apoya el aprendizaje de los contenidos del proceso de desarrollo de software**

**Figura 69.** Análisis Pregunta 10



La herramienta Uisport 2.0 es vista por la gran mayoría de los usuarios encuestados como un apoyo para el aprendizaje de la asignatura, como se observa en la figura 69. Sin embargo el 16% de los estudiantes a pesar de aceptar la herramienta, esperan un mejoramiento en este aspecto. Con solo 5% de Desacuerdo parcial, se puede afirmar que Uisport 2.0 cumple con uno de sus requerimientos más importantes: Apoyar el desarrollo de software.

Por último, la encuesta conto con una pregunta abierta, que sugería a los encuestados transmitir todas las observaciones y recomendaciones que surgieron durante la experiencia con la herramienta. Algunas de las más relevantes fueron:

- Notificar al usuario cuando se le ha compartido una evidencia.
- Al guardar un escenario, contar con la opción de crear otro sin volver al menú principal.
- Al momento de compartir un archivo pdf generado en la plataforma, los nombres no están diferenciados, se llaman de la misma manera.
- Algunos diagramas de casos de uso aparecen en el pdf, otros no.

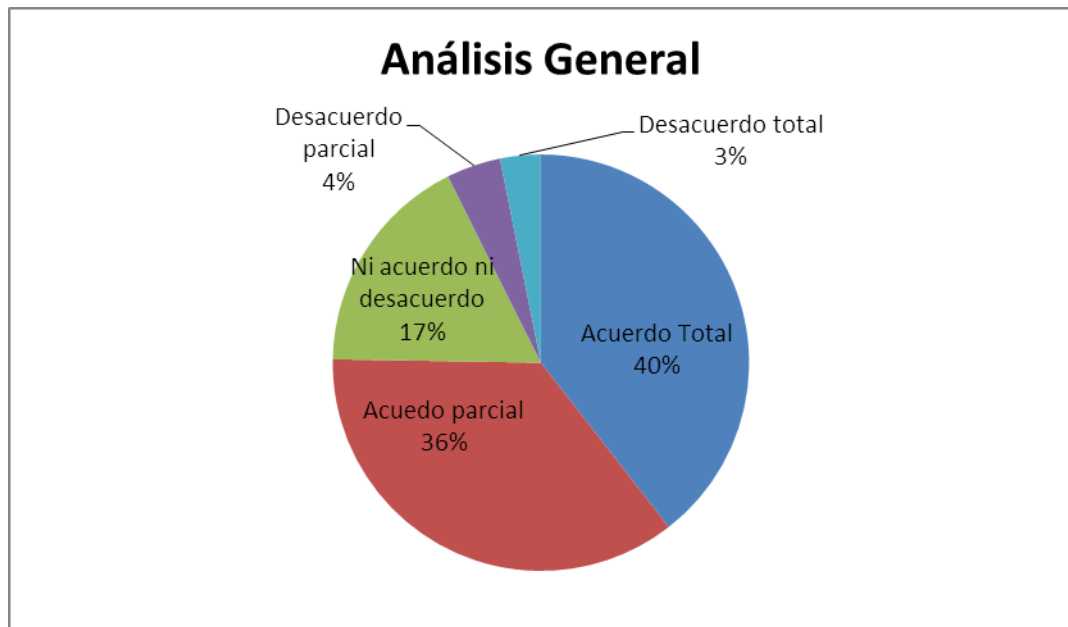
- Crear un botón que me permita editar el siguiente ítem, sin tener que volver al menú principal.

Para concluir el análisis de la Prueba piloto, y fundamentándolo en los resultados generales que pueden verse en la Figura 70, la herramienta conto con un grado alto de aceptación entre los estudiantes.

La mayoría de ellos quedaron satisfechos con la herramienta y entendieron rápidamente las instrucciones para su uso, prueba de ello es el 40% en Acuerdo total y el 36% en Acuerdo parcial.

Solo un 7% de los encuestados se mostró inconforme y no estuvo de acuerdo con algunas de las funcionalidades de la herramienta.

**Figura 70.** Análisis general de la prueba piloto



Teniendo en cuenta las observaciones y recomendaciones hechas por los estudiantes durante el desarrollo de la actividad, gran parte de estas fueron implementadas para la presentación final del software; otras se dejan como recomendación para el desarrollo de la siguiente versión de la herramienta *Uisport*.

## 6. CONCLUSIONES

- La metodología de Modelo cíclico mencionada por Pressman en su libro *Ingeniería del Software, Un enfoque práctico*, es una excelente opción cuando se analiza software del cual ya existe una versión, pues permite que el desarrollador se concentre en los cambios y realice el proceso de reingeniería de una forma adecuada.
- La reingeniería que se le aplicó a *Uisport 1.0* para obtener la nueva versión, tuvo como principal objetivo mejorar los procesos de la plataforma y de esta manera incrementar su calidad, estructura y por sobre todo aumentar la esperanza de vida del software, además de incluir nuevas funcionalidades para obtener un software confiable, eficiente, eficaz y efectivo.
- Fue posible realizar buenas prácticas de desarrollo de software con respecto a la documentación, ya que se contaba con una metodología apta para generar y dejar registro de los cambios implementados con el fin de facilitar el trabajo a futuros desarrolladores de la herramienta.
- Como producto final se definió un sistema consistente y funcional que integra diferentes herramientas implementadas en un espacio virtual de aprendizaje, el cual contribuye al proceso de enseñanza de los estudiantes de la asignatura Ingeniería del software I.
- La calidad del software debe primar sobre la cantidad de funcionalidades que ejecuta la aplicación. En la realización de las diferentes pruebas de software se evidenció que el usuario busca una herramienta que lo guíe y que funcione de manera correcta, sin importar la cantidad de usos o funciones con las que esta cuente.
- La realización de una prueba piloto es una elección correcta cuando se va a evaluar una herramienta pedagógica, ya que es muy importante que el usuario tenga contacto con el software antes de crear la versión final. Esto permite a los desarrolladores ver aspectos que solo el usuario final puede ver y por lo tanto pueden ser corregidos los errores o implementadas las recomendaciones para tener como resultado un software más completo y agradable.

- Con la realización de este proyecto, se pudo evidenciar la importancia que tienen las herramientas tecnológicas como apoyo en la enseñanza brindada por profesores y en su interacción con los estudiantes.
- La realización de este proyecto fue una experiencia enriquecedora tanto profesional como personalmente, pues se desarrollaron y perfeccionaron habilidades y conocimientos adquiridos durante el desarrollo de la carrera; permitiéndonos resolver problemas reales, y preparándonos para la vida profesional.

## 7. RECOMENDACIONES

- Tanto docentes como estudiantes de la asignatura Ingeniería de software I, deben hacer uso de la herramienta, pues es a través de sus observaciones y recomendaciones que se pueden implementar nuevas funcionalidades logrando una herramienta más completa.
- En busca de un mejoramiento futuro para la herramienta, tener en cuenta la construcción de un diagramador incorporado, para que el usuario pueda realizar los diagramas correspondientes a las etapas de desarrollo de software, sin tener que buscar herramientas externas y convertir dichos diagramas a imágenes para poder subirlas al bloque e-portafolio.
- Para versiones futuras, que la herramienta cuente con un banco de proyectos que se hayan realizado anteriormente con la misma. Esto con el fin de guiar al usuario en las distintas etapas que debe atravesar para el desarrollo de software.
- Aplicar la herramienta didáctica portafolio a otras asignaturas, ya que mediante esta es posible guiar al estudiante en el desarrollo de sus habilidades y permite estar en constante comunicación, para aportar el apoyo necesario.
- A los estudiantes no solo de Ingeniería de sistemas, si no de la Universidad Industrial de Santander, incentivar el desarrollo de herramientas pedagógicas que apoyen el aprendizaje de las diferentes asignaturas. Esto no solo beneficia al estudiante que usa la herramienta, si no que enriquece al ingeniero que la desarrolla, preparándolo para resolver problemas reales y enfrentarse a su vida profesional.

## BIBLIOGRAFIA

ACHOUR, Mehdi et al. Manual de PHP. [online] PHP Documentation Group [Barcelona: España] PHP.net. Disponible en internet: <http://php.net/manual/es/index.php>.

Bruegge, B., & Dutoit, A. (2010). Object-oriented Software Engineering (Tercera ed.). Mexico: Pearson Education.

Drake, José M. Ingeniería de Programación (4 Físicas). [online]. Disponible en internet: [http://www.ctr.unican.es/asignaturas/Ingenieria\\_Software\\_4\\_F/Doc/M0\\_08\\_Programa.pdf](http://www.ctr.unican.es/asignaturas/Ingenieria_Software_4_F/Doc/M0_08_Programa.pdf).

GALVIS PANQUEVA, Alvaro. Ingeniería de software educativo: Parte 2. Bogotá: Ediciones Uniandes, 1992.

Hammer, Reengineering Work: Don't Automate, Obliterate. En: Harvard Business Review, julio-agosto 1990, pp 104.

MOODLE, Acerca de moodle. [online]. Disponible en internet: <https://moodle.org/about/>

MYSQL. Mysql 5.0 Reference Manual. [online] Oracle and/or its affiliates. Mysql, 2011. Disponible en internet: <http://dev.mysql.com/doc/refman/5.0/es/index.html>.

Norma S. Cotic, E-learning: Una opción diferente para la capacitación docente. [online]. Disponible en internet: <http://www.soarem.org.ar/Documentos/25%20Cotic.pdf>

Pressman Roger. Ingeniería del software-Un enfoque práctico. Quinta edición. México: McGraw-Hill, 2002.

Rosenberg, M. (2001). E-learning: Estrategias para transmitir conocimiento en la era digital. Bogotá: McGrawHill

Ruiz, Nayibe. Estrategia didáctica virtual para apoyar el aprendizaje del concepto de representación en procesos de diseño de software. Bucaramanga, 2013, 128p. Trabajo de grado (Magister en ingeniería de sistemas e informática). Universidad industrial de Santander. Facultad de ingenierías físico-mecánicas.

Shulman, L. (1999). Portafolios del docente: una actividad teórica. En N. Lyons, N. (Comp.) (1999). El uso del portafolios. Propuestas para un nuevo profesionalismo docente. Buenos Aires: Amorrortu, 45-62.

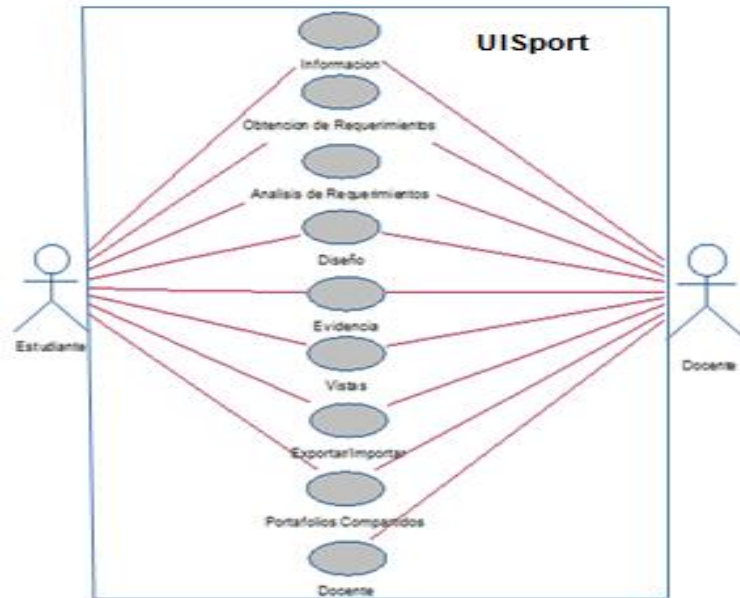
Sommerville, I. (2011). Ingeniería de software. Mexico: Pearson educación.  
URIBE TIRADO, Alejandro, MELGAR-ESTRADA, Liliana-María and BORNACELLY-CASTRO, Jaime-Alberto. "Utilización de Moodle en la gestión de información, documental y del conocimiento en grupos de investigación." El Profesional De La Información 16, no. 5 (Septiembre 2007): 468-474. Computers & Applied Sciences Complete, EBSCOhost .

[www.acis.org.co](http://www.acis.org.co) Recuperado el 31 de julio de 2014, de [http://www.acis.org.co/fileadmin/Base\\_de\\_Conocimiento/XXVII\\_Salon\\_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf](http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXVII_Salon_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf)



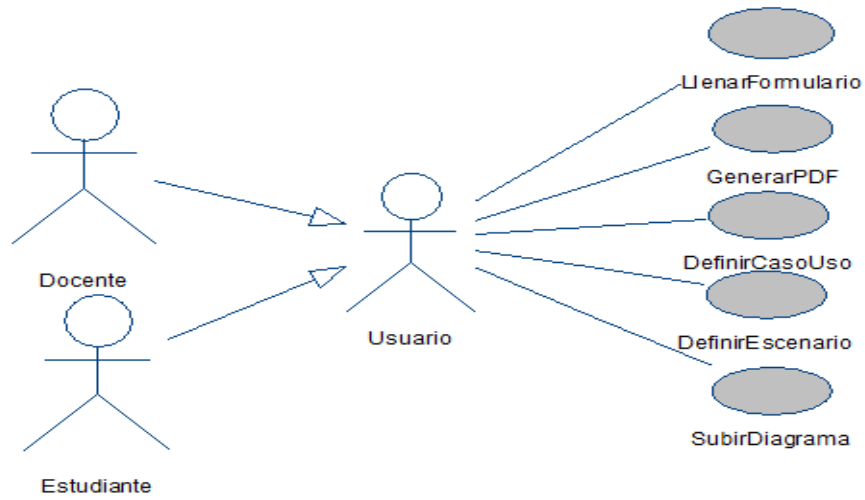
## DIAGRAMAS DE CASOS DE USO

Figura 72. Caso de uso: General



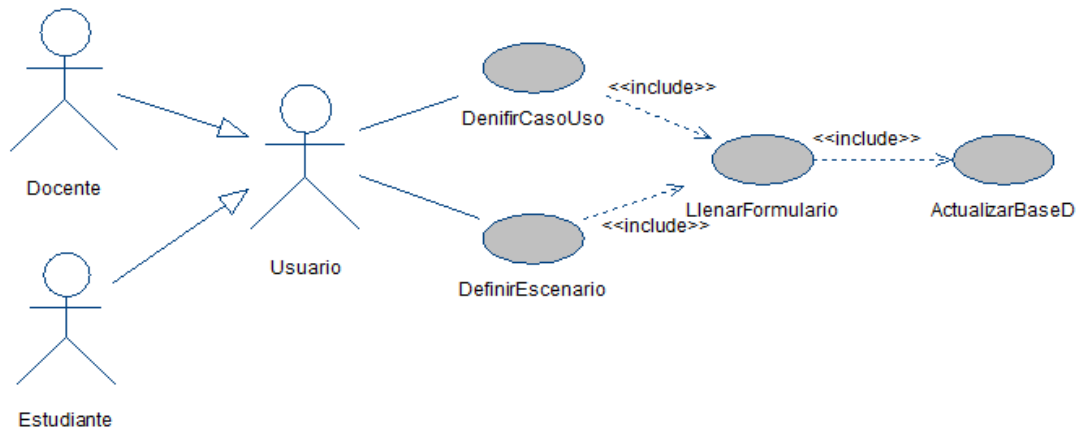
Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

**Figura 73. Caso de uso: ObtencionyAnalysis**



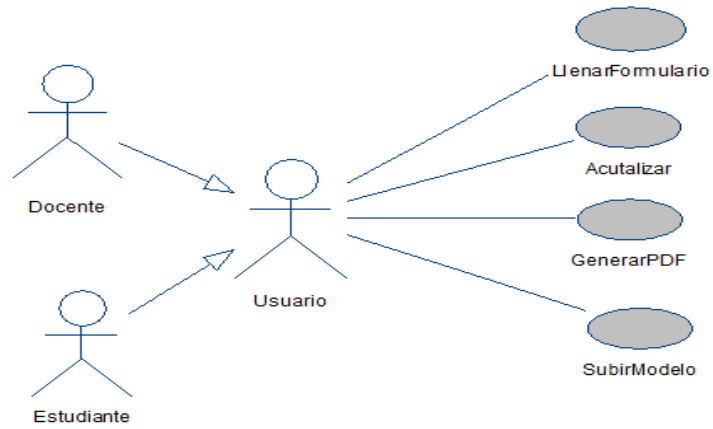
Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

**Figura 74. Caso de uso: DefinirCasoUso y DefinirEscenario**



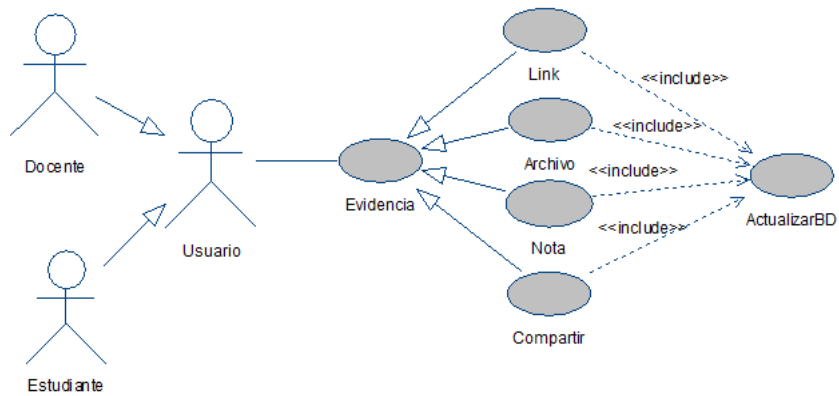
Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

**Figura 75. Caso de uso: Diseño**



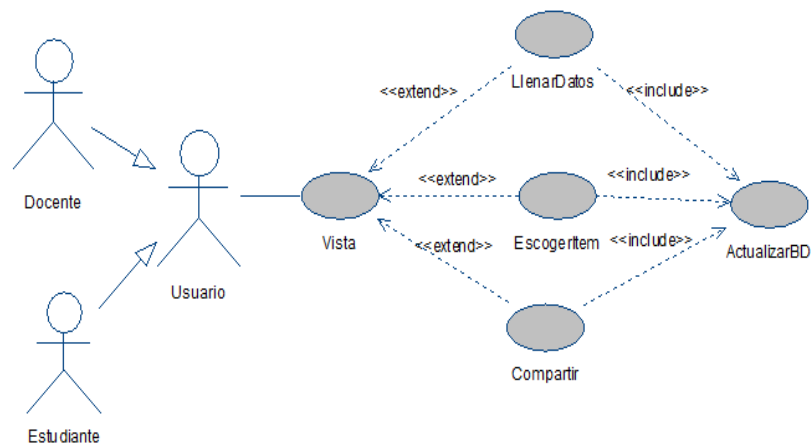
Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

**Figura 76. Caso de uso: Evidencia**



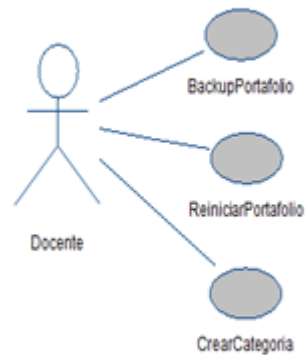
Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

**Figura 77. Caso de uso: Vista**



Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

**Figura 78. Caso de uso: Docente**



Fuente: Ruiz, N (2013). Estrategia didáctica para apoyar el aprendizaje del concepto de representación en proceso de diseño de software.

## DESCRIPCIÓN DE CASOS DE USO DE *Uisport*

**Tabla 2. Caso de uso: ObtencionyAnalysis**

Nombre del caso de uso	ObtencionyAnalysis
<b>Actor participante</b>	<b>Usuario:</b> Estudiante o Docente
<b>Condición inicial o de entrada</b>	El <b>usuario</b> ingresa al curso de Moodle y hace clic en el bloque Uisport
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El <b>usuario</b> hace clic en la pestaña de obtención de requerimientos.</li> <li>2. El <b>usuario</b> llena cada uno de los campos del formulario correspondiente a la etapa de análisis y diseño del sistema y hace clic en el botón guardar los cambios, también tiene la opción de regresar al menú y de esta manera trabajar en lo que desee.</li> <li>3. El <b>usuario</b> hace clic en la opción <b>Generar</b> y se genera un documento en formato .pdf teniendo la opción de guardarlo en un medio de almacenamiento.</li> <li>4. El <b>usuario</b> puede escoger el caso de uso <b>DefinirCasoUso</b> o <b>DefinirEscenario</b>.</li> <li>5. El <b>Usuario</b> hace clic en <b>SubirDiagrama</b> y le permite escoger el archivo que contenga el diagrama o modelo correspondiente al caso de uso.</li> <li>6. El <b>usuario</b> cuenta con las siguientes opciones: Abandonar la página y perder los cambios, o continuar editando y guardar los cambios. También tiene la opción de ir al siguiente ítem y así trabajar de forma ordenada siguiendo una secuencia de lo contrario puede elegir el botón de regresar al menú.</li> </ol>
<b>Condición de salida</b>	<b>Uisport</b> informa que se han guardado los ítems llenados en el formulario al

	<b>Usuario</b> y que el archivo .pdf ha sido generado con éxito.
<b>Requerimientos especiales</b>	El usuario debe estar registrado en el curso en el rol de estudiante o profesor.

**Tabla 3. Caso de uso: DefinirCasoUso**

Nombre del caso de uso	DefinirCasoUso o DefinirEscenario
<b>Actor participante</b>	<b>Usuario:</b> Estudiante o Docente.
<b>Condición inicial o de entrada</b>	El <b>Usuario</b> se encuentra dentro del bloque <i>Uisport</i> en la pestaña Obtención de requerimientos o Análisis de requerimientos.
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El <b>usuario</b> hace clic en DefinirCasoUso o DefinirEscenario.</li> <li>2. El <b>Usuario</b> llena cada uno de los campos del formulario correspondiente al caso de uso o escenario y hace clic en el botón <b>Guardar</b>. Además puede <b>actualizar</b> los campos e ingresar un nuevo escenario o volver al menú.</li> <li>3. El <b>usuario</b> hace clic en <b>Generar</b>, para que los datos queden actualizados en el archivo .pdf.</li> <li>4. El <b>usuario</b> después de crear un caso de uso tiene la opción de hacer clic en diagrama de caso de uso, esto lo llevara a la pestaña evidencias y podrá subir el diagrama correspondiente.</li> <li>5. El <b>usuario</b> también cuenta con la opción de crear un nuevo caso de uso.</li> </ol>
<b>Condición de salida</b>	<b>Uisport</b> informa que se han guardado los ítems llenados con éxito. Los datos son actualizados en el archivo .pdf generado.
<b>Requerimientos especiales</b>	El usuario debe estar registrado en el curso en rol de estudiante o profesor. La modificación o actualización de

Casos de uso y escenarios solo se puede hacer desde la pestaña obtención de requerimientos.

Tabla 4. Caso de uso: Diseño

Nombre del caso de uso	Diseño
<b>Actor participante</b>	<b>Usuario:</b> Estudiante o Docente
<b>Condición inicial o de entrada</b>	El <b>usuario</b> ingresa al curso de Moodle y hace clic en el bloque <i>Uisport</i> .
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El <b>usuario</b> hace clic en la pestaña de Diseño.</li> <li>2. El <b>usuario</b> llena cada uno de los campos del formulario de diseño hace clic en el botón <b>Guardar</b> los cambios lo que le mostrara el registro que se va a guardar, adicional a esto puede <b>Actualizar</b> el registro o simplemente volver al menú.</li> <li>3. El <b>usuario</b> hace clic en la opción <b>Generar</b> y se genera un documento en formato .pdf teniendo como opción guardarlo en un medio de almacenamiento.</li> <li>4. El <b>usuario</b> hace clic en <b>SubirModelo</b> y le permite escoger el archivo que contenga el diagrama o modelo correspondiente a la etapa de diseño.</li> </ol>
<b>Condición de salida</b>	<i>Uisport</i> informa que se han guardado los ítems llenados del formulario al <b>Usuario</b> , que el archivo .pdf ha sido generado con éxito y los diagramas o modelos aparecen en la pestaña registrados.
<b>Requerimientos especiales</b>	El usuario debe estar registrado en el curso en el rol de estudiante o profesor.

Tabla 5. Caso de uso: Evidencia

Nombre del caso de uso	Evidencia
<b>Actor participante</b>	<b>Usuario:</b> Estudiante o Docente
<b>Condición inicial o de entrada</b>	El <b>usuario</b> ingresa al curso de Moodle y hace clic en el bloque <i>Uisport</i> .
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El <b>usuario</b> hace clic en la pestaña de Evidencia.</li> <li>2. El <b>usuario</b> escoge una de las tres opciones de evidencia: link, archivo o nota.</li> <li>3. El <b>usuario</b> llena los campos correspondientes al nombre, categoría o tipo y selecciona el archivo a subir en a la plataforma.</li> <li>4. El <b>usuario</b> hace clic en <b>guardar cambios</b>.</li> <li>5. El <b>usuario</b> puede elegir si quiere compartir la evidencia que acaba de crear.</li> <li>6. El <b>usuario</b> puede ver las imágenes que ha subido y también tiene la opción de <b>descargar</b> cualquier archivo que suba o comparta.</li> </ol>
<b>Condición de salida</b>	<i>Uisport</i> informa que se han subido las evidencias con éxito y muestra en la pantalla la evidencia con la fecha, categoría y las opciones de modificar, eliminar o compartir.
<b>Requerimientos especiales</b>	El usuario debe estar registrado en el curso en el rol de estudiante o profesor. Solo el creador de la evidencia puede eliminar la misma.

Tabla 6. Caso de uso: Vista

Nombre del caso de uso	Vista
<b>Actor participante</b>	<b>Usuario:</b> Estudiante o Docente
<b>Condición inicial o de entrada</b>	El <b>usuario</b> ingresa al curso de Moodle y hace clic en el bloque <i>Uisport</i> .
<b>Flujo de eventos</b>	1. El <b>usuario</b> hace clic en la

	<p>pestaña evidencias.</p> <ol style="list-style-type: none"> <li>2. El <b>usuario</b> hace clic en <b>Agregar Vista</b>.</li> <li>3. El <b>usuario</b> llena los campos correspondientes al nombre de la vista y selecciona las evidencias o elementos que desea que contengan la vista.</li> <li>4. El <b>usuario</b> hace clic en <b>compartir</b> y <i>Uisport</i> le muestra los usuarios de Moodle con los cuales puede compartir vista.</li> <li>5. El <b>usuario</b> hace clic en guardar cambios.</li> </ol>
<b>Condición de salida</b>	<i>Uisport</i> informa que se ha creado la vista con éxito y muestra en la pantalla datos, fecha de creación de la vista y, los usuarios que tienen acceso a ella y la opción de modificar o eliminar.
<b>Requerimientos especiales</b>	El usuario debe estar registrado en el curso en el rol de estudiante o profesor. Solo el creador de la vista puede eliminar la misma.

**Tabla 7. Caso de uso: Docente**

Nombre del caso de uso	Docente
<b>Actor participante</b>	<b>Docente</b>
<b>Condición inicial o de entrada</b>	El <b>Docente</b> ingresa al curso de Moodle y hace clic en el bloque <i>Uisport</i> .
<b>Flujo de eventos</b>	<ol style="list-style-type: none"> <li>1. El <b>docente</b> hace clic en la pestaña Docente.</li> <li>2. El <b>docente</b> observa el listado de los estudiantes, el grupo correspondiente y las evidencias que haya registrado.</li> <li>3. El <b>docente</b> puede escoger la opción <b>Generar</b> y con ello realiza una copia de seguridad de todos los portafolios de los estudiantes, en un archivo en .pdf.</li> <li>4. El docente escoge la opción <b>Editar</b>, en cuyo caso puede</li> </ol>

	<p>agregar o modificar una categoría para los diferentes tipos de evidencias a manejar, <i>Uisport</i> le informa que los cambios se han realizado con éxito.</p> <p>5. El docente escoge la opción <b>Reiniciar</b> y con ello elimina todos los registros del portafolio actual con el fin de iniciar un nuevo curso, <i>Uisport</i> le pide confirmación de la acción.</p>
<b>Condición de salida</b>	<ul style="list-style-type: none"> <li>• Uisport informa que se ha creado con éxito el archivo en .pdf de la copia de todos los portafolios.</li> <li>• Muestra en pantalla la nueva categoría o los cambios realizados a estas.</li> <li>• Muestra el mensaje de eliminación de todos los portafolios.</li> </ul>
<b>Requerimientos especiales</b>	El docente debe estar registrado en el curso en el rol de profesor y solo el puede ver esta pestaña.

## **Anexo B. MANUAL DE USUARIO**

**Reingeniería de la aplicación *Uisport* que soporta la enseñanza de procesos de representación en ingeniería del software.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERIAS FISICO-MECANICAS  
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA  
BUCARAMANGA  
2014**

# MANUAL DE USUARIO

## INTRODUCCIÓN

El objetivo de este apartado es el de explicar el funcionamiento de la aplicación *Uisport* que soporta la enseñanza de procesos de representación en ingeniería del software.

El usuario mediante esta aplicación podrá realizar diferentes procesos segmentados por etapas que contribuyen a crear el documento RAD y otro tipo de información necesaria en un desarrollo de software.

## ACCESO A LA APLICACIÓN

El acceso a la aplicación se realizará mediante el empleo de un navegador WEB.

Cabe destacar que esta aplicación es compatible con “Internet Explorer 8”, “Mozilla Firefox” y “Google Chrome”.

La herramienta se desarrolló para la plataforma Moodle a manera de bloque, para ser incorporado en uno o varios cursos. En primer lugar ingresamos a la plataforma en que se encuentre dicha herramienta. Actualmente la aplicación se encuentra instalada en la plataforma <http://tic.uis.edu.co>.

Una vez que hayamos accedido a la aplicación nos aparecerá la siguiente pantalla:

**Figura 79. Pantalla de ingreso**



The screenshot shows a login form with the following elements:

- Header: **Entrar**
- Form fields: "Nombre de usuario" and "Contraseña", each with an adjacent input box.
- Submit button: "Entrar" (grey button).
- Checkbox: "Recordar nombre de usuario" with an unchecked box.
- Link: "¿Olvidó su nombre de usuario o contraseña?" in blue text.
- Footer: "Las 'Cookies' deben estar habilitadas en su navegador" with a help icon.

La función de esta pantalla es distinguir tres niveles de acceso: Administrador, Docente y estudiante. Cada nivel de acceso tendrá asignadas unas funcionalidades distintas, determinadas por Moodle y por el administrador del sitio. Para poder acceder a la plataforma el usuario debe contar con una cuenta previamente creada, y en el caso del docente y el estudiante, estar matriculados en la asignatura que cuenta con el bloque e-portafolio.

Después de ingresar a su cuenta, el usuario debe dar clic en el curso en que se encuentra instalado el bloque e-portafolio.

Los pasos para agregar el bloque, se encuentran en el anexo 5. *Bloque e-portafolio*.

**Figura 79. Bloque para docente**



Si el usuario se ha identificado como docente se encontrara con un bloque como el que se muestra en la Figura 2. El docente podrá realizar las siguientes actividades en el bloque Uisport:

- El usuario puede hacer clic en la pestaña Docente.
- Puede observar el listado de los estudiantes, el grupo correspondiente y las evidencias que haya registrado.
- Puede escoger la opción Generar y con ello realizar una copia de seguridad de todos los portafolios de los estudiantes, en un archivo en .PDF.
- Puede escoger la opción Editar, en cuyo caso puede agregar o modificar una categoría para los diferentes tipos de evidencias a manejar.
- Puede escoger la opción Reiniciar y con ello elimina todos los registros del portafolio actual con el fin de iniciar un nuevo curso.
- Puede usar cada una de las pestañas de la herramienta, las cuales se identifican en la Figura 4.

**Figura 80. Bloque para estudiante.**



Si el usuario se ha identificado como estudiante se encontrara con un bloque como el que se muestra en la Figura 3. El estudiante podrá hacer uso de cada una de las pestañas mostradas en la Figura 4.

A continuación indicaremos las diferentes pestañas con las que cuenta el bloque e-portafolio.

- |                                      |                                  |
|--------------------------------------|----------------------------------|
| <b>1</b> Información                 | <b>5</b> Evidencias              |
| <b>2</b> Obtención de requerimientos | <b>6</b> Vistas                  |
| <b>3</b> Análisis de requerimientos  | <b>7</b> Exportar/importar       |
| <b>4</b> Diseño                      | <b>8</b> Portafolios Compartidos |

**Figura 81. Descripción pantalla principal**



## 1. INFORMACIÓN:

Descripción de la información personal del usuario

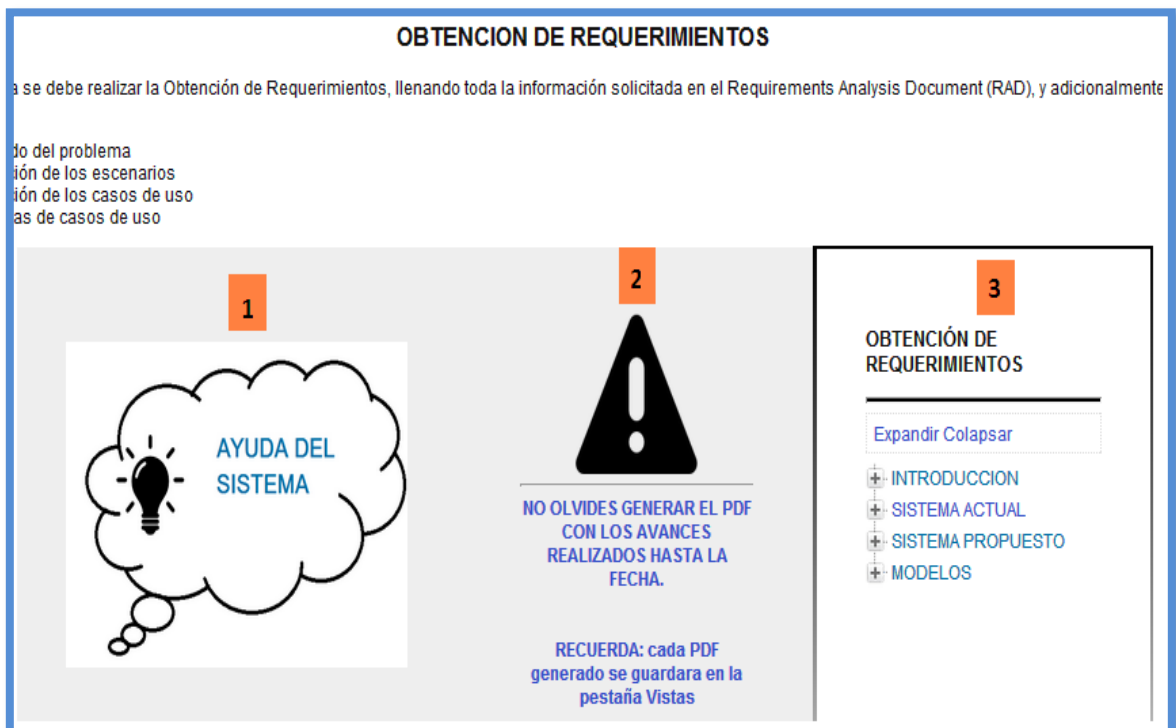
**Figura 82. Información**



## 2. OBTENCIÓN DE REQUERIMIENTOS:

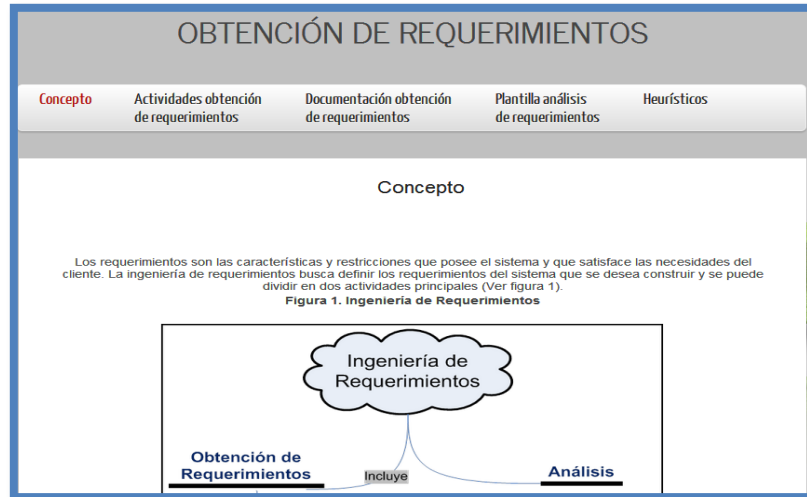
En esta pestaña el usuario puede ingresar la información correspondiente con el RAD para la creación de un proyecto software.

**Figura 83. Pestaña obtención de requerimientos**



- 1) Ayuda del sistema: Permite que el usuario se apoye en los heurísticos para realizar las actividades correspondientes. Esta opción también está disponible en las pestañas 3. Análisis de requerimientos y 4. Diseño

**Figura 84. Heurístico Obtención de requerimientos**

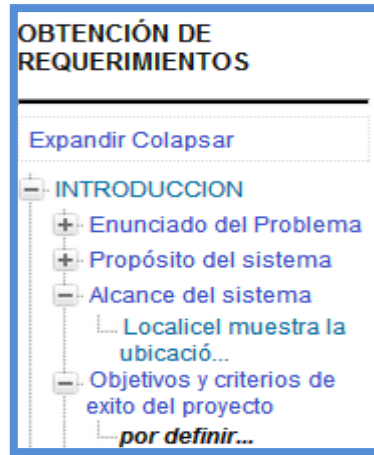


- 2) Generar PDF: Después de haber registrado la información sugerida por la plataforma, esta opción genera el RAD con el fin de que el usuario vea el compilado de su proyecto.

**Figura 85. PDF**

- 3) Menú obtención de requerimientos: a través de este menú, el usuario encuentra cada uno de los requisitos que debe diligenciar, para completar el documento RAD. Además puede visualizar que información ha ingresado y cual le falta *por definir*.

**Figura 86. Menú obtención de requerimientos**



- Después de ingresar la información sugerida y guardar los cambios se cuenta con dos opciones:
  - ✓ Dar clic en el botón siguiente ítem, el cual le permite ingresar la información correspondiente a la pestaña en que se encuentra.
  - ✓ Dar clic en el botón volver al menú principal, que lo envía al menú de la pestaña en que se encuentra.

Cabe resaltar que estas 3 funcionalidades mencionadas anteriormente, se repiten en las pestañas: 3. Análisis de Requerimientos y 4. Diseño.

**Figura 87. Opciones editar-volver**



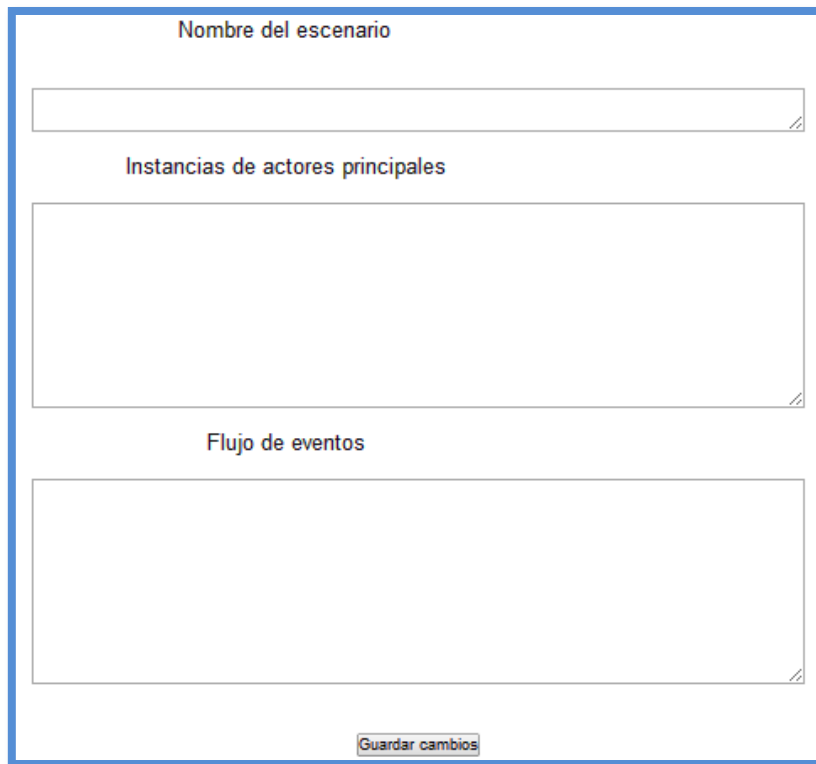
- El menú desplegable cuenta con un ítem llamado MODELOS que permite ingresar toda la información propia de los modelos necesarios en la etapa Obtención de requerimientos.  
Hay que tener en cuenta que esta opción también se encuentra disponible en las pestañas 3.Análisis de requerimientos y 4.Diseño; descritas más adelante.

**Figura 88. Modelos**



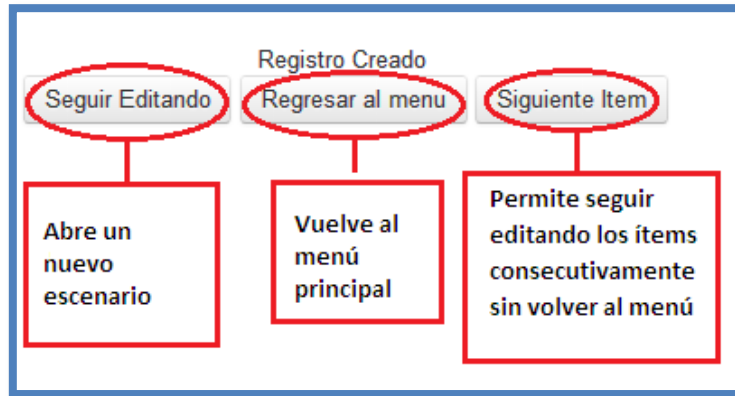
- Escenarios: permite ingresar el paso a paso de un escenario.

**Figura 89. Escenarios**

A screenshot of a form for creating a scenario. The form is enclosed in a blue border and contains three main sections, each with a text input field: 'Nombre del escenario', 'Instancias de actores principales', and 'Flujo de eventos'. At the bottom of the form, there is a button labeled 'Guardar cambios'.

Después de guardar los cambios, la plataforma ofrece tres opciones:

**Figura 90. Escenario guardado-opciones**



- Casos de uso: permite ingresar los diferentes casos de uso paso a paso.

**Figura 91. Ingresar caso de uso**

Nombre del caso de uso

Actores participantes

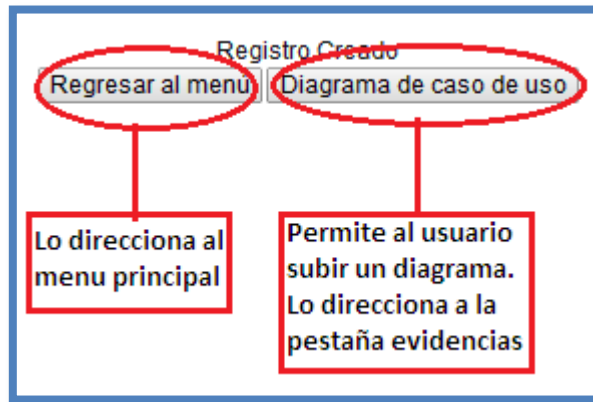
Flujo de eventos

Condiciones de entrada

Condiciones de salida

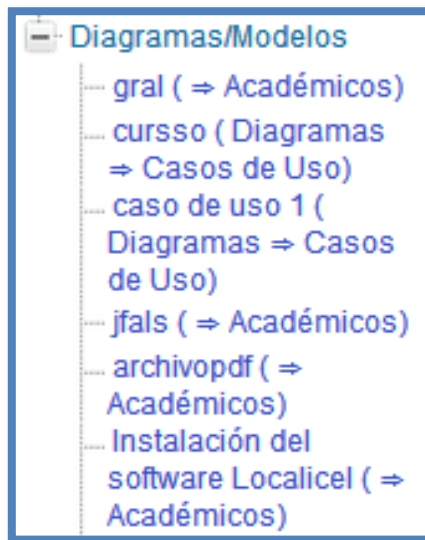
Después de guardar los cambios, se cuentan con tres opciones:

**Figura 92. Guardar caso de uso-opciones**



- Diagramas/modelos: permite visualizar los diagramas que se han subido a la pestaña evidencias.

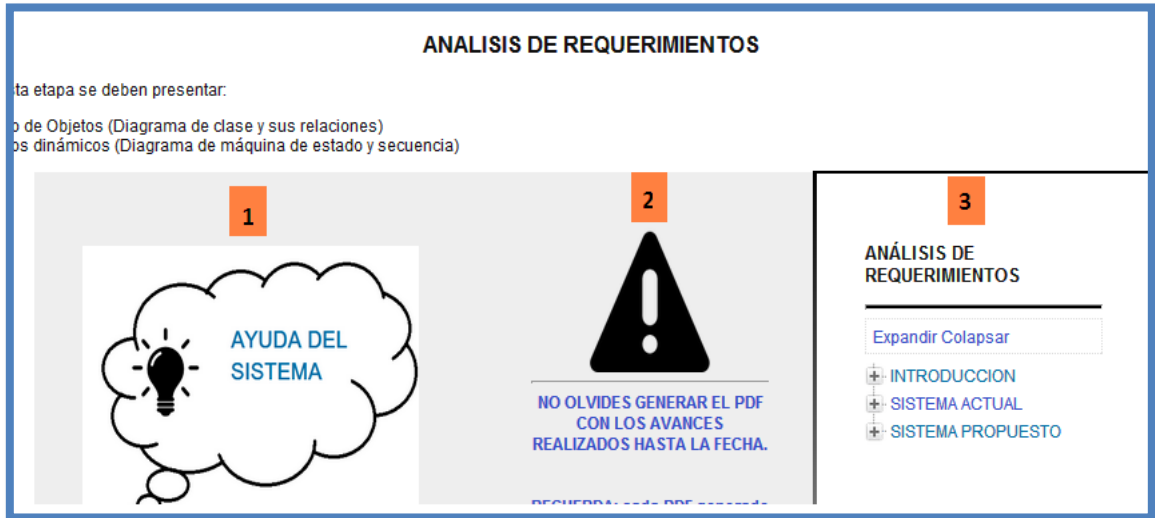
**Figura 93. Diagramas/modelos**



### 3. ANÁLISIS DE REQUERIMIENTOS:

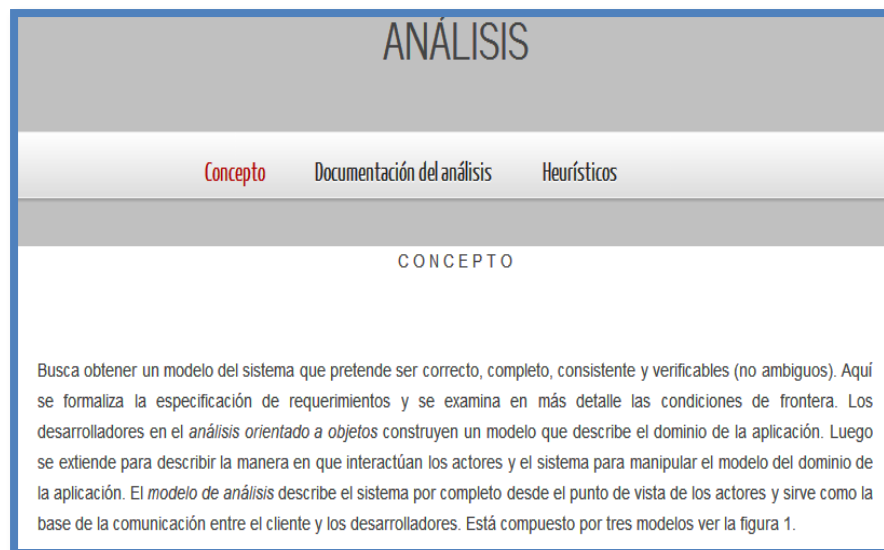
Esta pestaña está enfocada al análisis del documento RAD, cuya información fue ingresada en la pestaña 2. Obtención de requerimientos.

**Figura 94. Análisis de requerimientos**



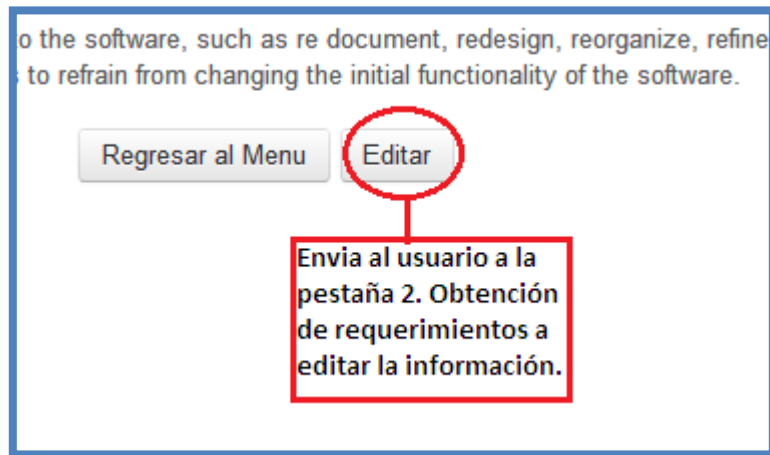
- 1) Ayuda del sistema: Como se mencionó en la pestaña 2. Obtención de requerimientos, esta opción le proporciona marco teórico al usuario sobre la pestaña que está trabajando.

**Figura 95. Heurístico Análisis de requerimientos**



- 2) Generar PDF: cumple la misma función que en la pestaña 2. Obtención de requerimientos. (ver Figura 85)
- 3) Menú Análisis de requerimientos: Es un menú desplegable que permite ver la información ingresada en la pestaña Obtención de requerimientos. Si el usuario desea editar esta información cuenta con esa opción:

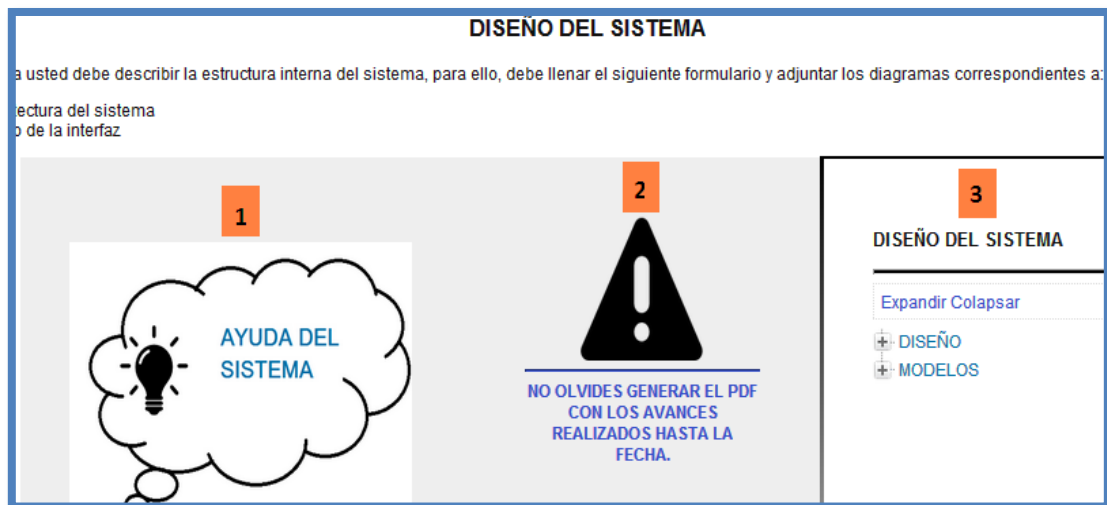
**Figura 96. Opción editar- Análisis de requerimientos**



#### 4. DISEÑO:

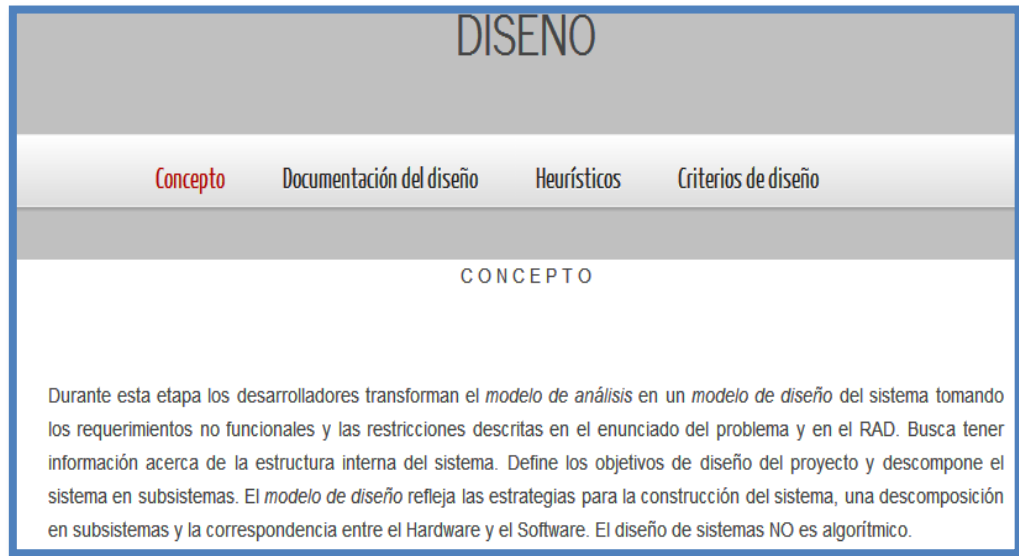
En esta pestaña el usuario ingresa toda la información referente al diseño del desarrollo del software.

**Figura 97. Diseño**



- 1) Ayuda del sistema: Marco teórico sobre el diseño del sistema.

**Figura 98. Heurístico Diseño**



- 2) Generar PDF; (ver Figura 85). Su función se describe en la pestaña 2.Obtencion de requerimientos.
- 3) Menú Diseño: Menú desplegable que permite ingresar información relacionada con el diseño del software que se está desarrollando.

**Figura 99. Menú Diseño**



5. **EVIDENCIAS:** Esta pestaña se creó con el fin de que el usuario pueda subir al sistema todo tipo de información que crea relevante para el desarrollo del software.

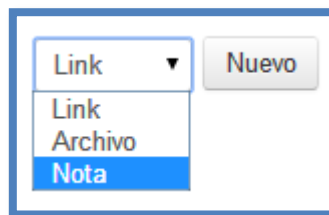
**Figura 100. Evidencias**

Ordenar por: <a href="#">Todos los Items</a>   <a href="#">Links</a>   <a href="#">Archivos</a>   <a href="#">Notas</a>					
Todos los artículos en su portafolio					
Categoría	Tipo	Nombre	Fecha ▾	Curso	Comen.
Diagramas => Casos de Uso	Archivo	general	domingo, 19 de octubre de 2014, 19:52	UISPORT pro	0

En la pestaña evidencias, se puede ver el listado de las evidencias subidas por el usuario, y también puede ordenarlas de acuerdo a su preferencia.

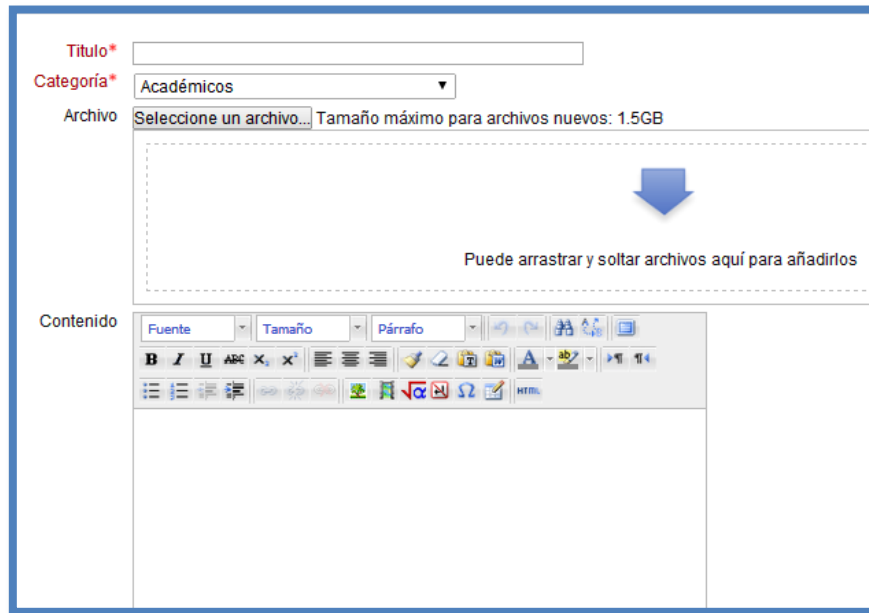
- Si el usuario desea crear una nueva evidencia debe seguir estos pasos:
  - a) Debe seleccionar el tipo de archivo que va a subir a la plataforma, y dar clic en Nuevo:

**Figura 101. Nueva evidencia**



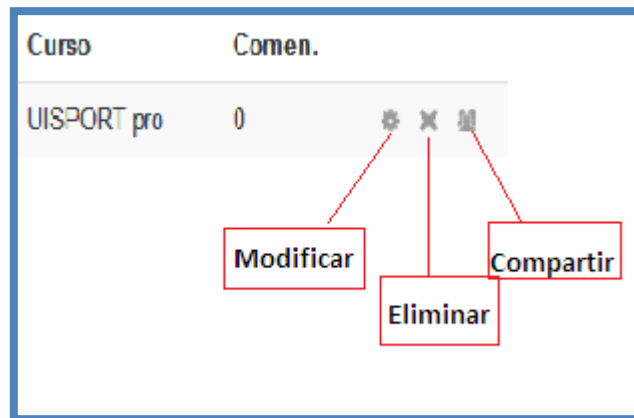
- b) El usuario debe poner un título que identifique la evidencia, escoger la categoría a la que corresponde el archivo y subirlo a la plataforma. Describir la evidencia es opcional, el usuario debe decidir si la evidencia requiere algún tipo de aclaración o no (ver figura 102).
- La pestaña Evidencias también ofrece las opciones editar, compartir o eliminar. Justo al lado derecho, está la opción *comen.* que muestra la cantidad de comentarios que los usuarios que han recibido la evidencia compartida, pueden dejar. (Ver figura 103).

**Figura 102. Subir una evidencia**



Formulario para subir una evidencia. Incluye campos para Título\*, Categoría\* (Académicos), Archivo (Seleccione un archivo... Tamaño máximo para archivos nuevos: 1.5GB) y Contenido (editor de texto con barra de herramientas).

**Figura 103. Evidencias-opciones**



- ✓ Modificar: permite cambiar el título, la descripción o la categoría de la evidencia.
- ✓ Eliminar: Permite eliminar del sistema la evidencia subida.
- ✓ Compartir: Permite compartir con cualquier otro usuario la evidencia. La herramienta Uisport direcciona al usuario a la pestaña vistas, la cual será descrita a continuación.

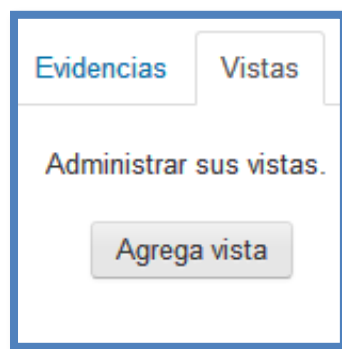
## 6. VISTAS:

Esta pestaña fue especialmente creada con el fin de que el usuario pueda compartir sus evidencias con cualquier usuario que se encuentre matriculado en la plataforma.

La función principal de esta pestaña consiste en que el usuario pueda trabajar en equipo, compartiendo sus archivos con compañeros de curso, o haciendo entregas constantes de su proyecto al docente encargado.

- Los pasos para compartir una evidencia son:
  - a) El usuario puede compartir evidencias desde la pestaña Evidencias como se mostró en la figura 103; o desde la pestaña vistas, así:

**Figura 104. Compartir evidencia desde vistas**



- b) Después el usuario debe ingresar un título que es obligatorio, y una descripción que es opcional.

En la opción “sus ítems” dar clic sostenido sobre la evidencia que desea compartir, y la arrastra hasta *diseño de vista*. (Ver figura 105)

- c) Por ultimo escoge los usuarios a quienes desea compartir la evidencia.

Las opciones son:

- Acceso Externo: usuarios ajenos a su curso.
- Acceso interno: compañeros de curso, así:
  - ✓ Acceso de todos
  - ✓ Acceso solo para los siguientes usuarios: escoge los usuarios con los que compartirá la evidencia, de entre sus compañeros y docentes de curso.

En esta opción el estudiante puede ver quienes están matriculados como estudiantes, y quienes como docentes del curso. (Ver figura 106)

**Figura 105. Ingresar vista**

The screenshot shows a web form titled 'Vista:'. Below the title is a section 'Información de la vista'. It contains two main input areas: 'Titulo\*' with a text box, and 'Descripción' with a larger text area. At the bottom, there are two buttons: 'Sus Items' (with a folder icon) and 'Diseño de Vista' (with a palette icon). Below 'Sus Items' is a sub-section with a 'general' link.

Los usuarios que reciben la vista pueden comentarla, y esto aparecerá en la pestaña evidencias, junto a las opciones, modificar, eliminar o compartir.

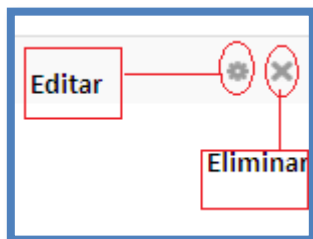
**Figura 106. Compartir evidencia-usuarios**

The screenshot shows a 'Compartir' dialog box. The title bar reads 'Compartir: Acceso Interno: Acceso sólo para los siguiente usuarios'. There are two main options: 'Acceso Externo' (unchecked) and 'Acceso Interno' (checked). Under 'Acceso Interno', there are two radio buttons: 'Acceso de todos' (unselected) and 'Acceso sólo para los siguiente usuarios' (selected). Below this, the text 'INGENIERIA DEL SOFTWARE I : 2014-2-22969-B1' is displayed. A table lists users with checkboxes for sharing and their roles.

	Nombre	Role
<input type="checkbox"/>	Sonia Cristina Gamboa Sarmiento	Profesor
<input type="checkbox"/>	MARIA FERNANDA FUENTES MANTILLA	Profesor
<input type="checkbox"/>	Ronald Augusto Chaparro Medina	Estudiante
<input type="checkbox"/>	Maria Juliana Garzon Vargas	Estudiante

- La pestaña vista, también ofrece la opción de ver la lista de vistas y modificarlas o eliminarlas, así:


**Figura 107. Vistas-opciones**



## 7. PORTAFOLIOS COMPARTIDOS:

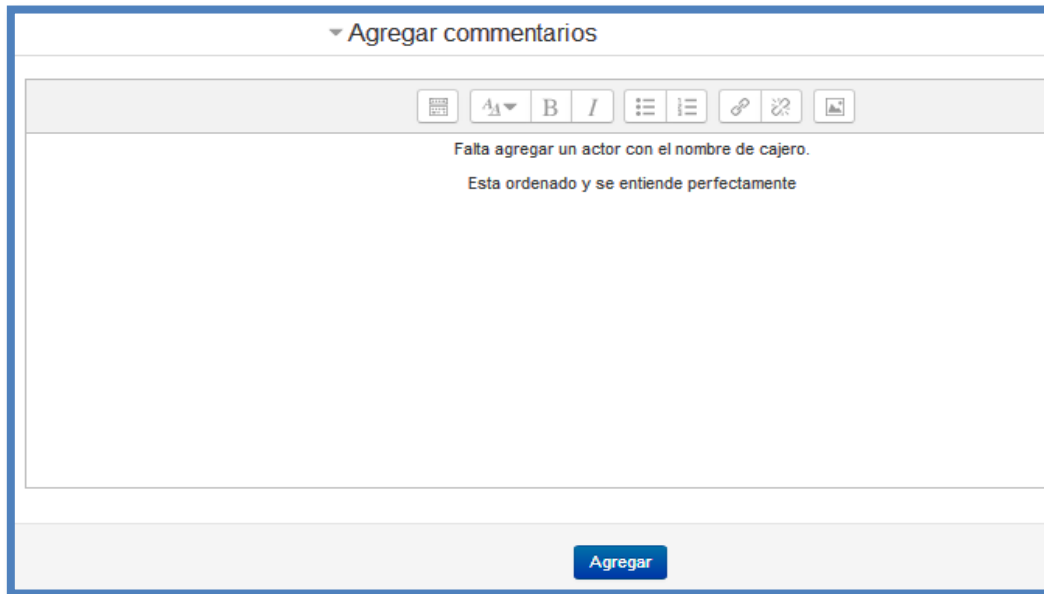
En esta pestaña el usuario puede visualizar todas las evidencias que otros usuarios le han compartido. El archivo tendrá el nombre que el usuario emisor le asigno.

**Figura 108. Portafolios compartidos**

Ordenar por: <a href="#">Curso</a>   <a href="#">Usuario</a>   <a href="#">Vista</a>   <a href="#">Fecha</a>			
Imagen del usuario	Usuario ▲	Vista	Fecha
	Fabio gutierrez	<a href="#">GENERAL</a>	domingo, 19 de octubre de 2014, 21:50

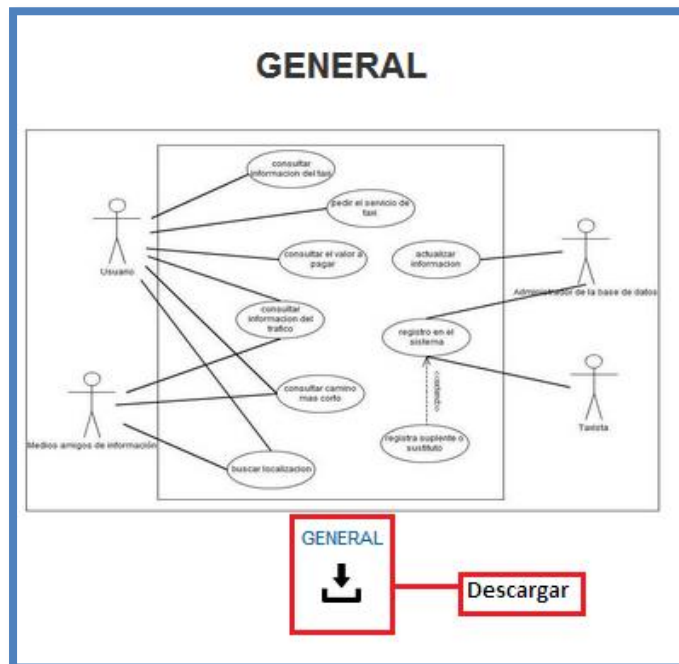
- El usuario receptor, puede ver la evidencia compartida y puede dejar comentarios para que el usuario emisor los revise. Estos comentarios aparecerán en la pestaña evidencias del usuario emisor.  
Cuando el usuario receptor da clic sobre el portafolio compartido, la plataforma lo direcciona a la pestaña vistas.

**Figura 109. Comentarios en la evidencia compartida**



- Adicional a esto, cuando el usuario emisor recibe una imagen o diagrama, tiene la opción de verla sin necesidad de descargarla. Pero si desea descargarla también cuenta con esta opción.

**Figura 110. Descargar portafolio compartido**



## **Anexo C. MANUAL TECNICO**

### **PLATAFORMA UISPORT 2.0**

Los cambios realizados en la plataforma a nivel de código serán descritos en este capítulo.

Es importante tener en cuenta las siguientes indicaciones a la hora de instalar *Uisport* o actualizarla, para no tener problemas con los strings:

1. Ingresar a la plataforma donde este instalada, o se vaya a instalar la herramienta Uisport, como administrador. En este caso al portal <http://tic.uis.edu.co>.
2. ingresar a: Administración del sitio/Extensiones/Bloques/Gestionar bloques
3. Buscamos el bloque Uisport y lo desinstalamos.
4. Cuando nos pregunta si queremos eliminar la carpeta Uisport, le decimos cancelar.
5. Le damos continuar para actualizar la base de datos de moodle.
6. Reemplazamos la antigua carpeta Uisport por la nueva, en el servidor.
7. Para instalar la nueva versión de la carpeta Uisport, es necesario entrar al curso al que se le desea instalar el bloque e-portafolio.
8. En la opción ADMINISTRACION damos clic sobre activar edición
9. Se cargara un menú a manera de bloque, que dice: AGREGAR BLOQUE.
10. Desplegamos el menú y escogemos la opción e-portafolio.
11. Volvemos a la opción ADMINISTRACION y damos clic en desactivar edición.
12. Si el bloque se instaló correctamente, aparecerá a manera de bloque cada vez que ingresemos al curso.
13. Por último, dar clic en Administración del sitio/ Idioma/Ajustes de idioma
14. Buscar la opción Guardar en cache todas las cadenas de Idioma y verificar que no esté seleccionado, si lo está, quitarlo, y guardo los cambios. Esto con el fin de actualizar las cadenas de idioma.
15. Después de verificar que la herramienta está funcionando correctamente, volver a el paso 13 y 14, y seleccionar la casilla Guardar en cache todas las cadenas de Idioma. Después guardar los cambios.

Si la carpeta Uisport va a ser instalada por primera vez, siga los pasos desde el 7 en adelante, de lo contrario, si va a actualizar la carpeta, sígalos todos.

A continuación serán descritos los archivos correspondientes a las funcionalidades de Uisport 2.0, los cuales fueron sometidos a modificaciones para la realización de la reingeniería; y para el mejoramiento de la plataforma.

## ARCHIVOS MODIFICADOS

Los archivos modificados en las diferentes pestañas son los siguientes:

**Tabla 8. Archivos modificados**

Nombre	Descripción
<b>Block_Uisport.php</b>	Contiene la definición de algunas clases y es el archivo principal del bloque
<b>Check_db_syntax.php</b>	Revisa si las sentencias sql están bien escritas
<b>Edit.php</b>	Genera el cuadro de texto y la opción de guardar al ingresar el contenido en obtención, y diseño excepto en casos de uso y escenarios
<b>Editcas.php</b>	Genera la interfaz para ingresar un caso de uso
<b>Editesc.php</b>	Genera la interfaz para ingresar un escenario
<b>Exportimport.php</b>	Genera la interfaz en la pestaña importar/exportar
<b>Genftp.php</b>	Genera el archivo pdf usando la librería fpdf.php
<b>Import_moodle, import_file.php</b>	Se usan desde el menú importer/exporter para importar archivos
<b>Inc.php</b>	Librería para incluir archivos
<b>Item.php</b>	Interfaz para subir una evidencia
<b>Portfoliofile.php</b>	Genera el link para descargar una evidencia
<b>Restart.pf</b>	Reinicia la plataforma
<b>Shared_views.php</b>	Interfaz en la pestaña portafolios compartidos
<b>Versión.php</b>	Archivo que describe la versión y la fecha de modificación del bloque
<b>View.php</b>	Interfaz de la pestaña información
<b>View_analysis.php</b>	Pestaña análisis
<b>View_categories.php</b>	Pestaña docente
<b>View_design.php</b>	Pestaña diseño
<b>View_items.php</b>	Pestaña evidencias
<b>View_obtention.php</b>	Pestaña obtención
<b>View_list.php</b>	Pestaña vistas

<b>Views_mod.php</b>	Agregar una nueva vista
----------------------	-------------------------

**Tabla 9. CARPETAS**

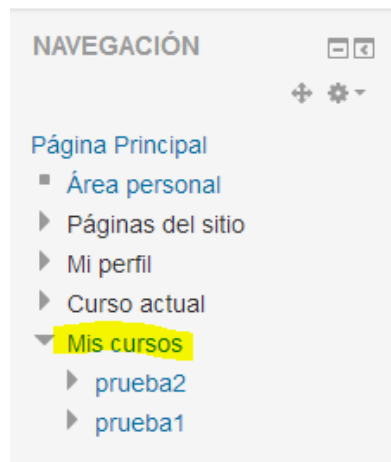
<b>Carpeta</b>	<b>Descripción</b>
<b>Es</b>	Contiene las cadenas en español, para cuando se escoge idioma español en la plataforma
<b>esMx</b>	Contiene las cadenas en español, para cuando se escoge idioma español México en la plataforma
<b>En</b>	Contiene las cadenas en inglés para cuando se escoge idioma ingles en la plataforma
<b>Js y javascript</b>	Tiene los archivos javascript para manejar los menús
<b>Db</b>	Contiene los archivos para crear la base de datos al momento de instalar el bloque.

## Anexo D. BLOQUE E-PORTAFOLIO

El administrador de la plataforma en que se va a instalar la herramienta Uisport 2.0, debe seguir estos pasos para agregar el bloque e-portafolio:

1. Ingresar a moodle (en este caso a <http://tic..uis.edu.co>) como profesor o como administrador del sitio.
2. En la parte superior izquierda, en la pestaña NAVEGACION- mis cursos, escoger el curso al que se desea activarle el bloque:

**Figura 111. Navegación**



3. En la pestaña administración, le doy clic a Activar Edición:

**Figura 111. Activar edición**



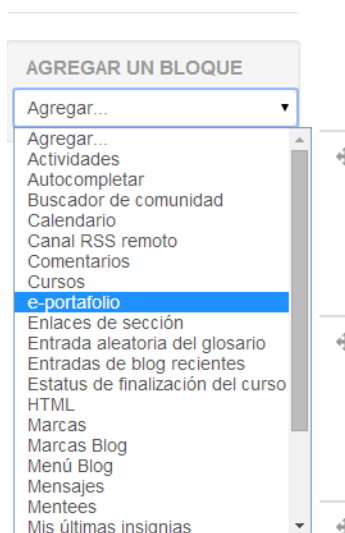
4. En la parte inferior izquierda de la ventana aparece la pestaña AGREGAR UN BLOQUE

**Figura 112. Agregar bloque**



5. Escogemos la opción e-portafolio

**Figura 113. e-portafolio**



6. Cada vez que ingrese al curso tanto el profesor, como cualquier estudiante matriculado en este, en la parte inferior izquierda (depende de la versión de moodle) aparecerá el bloque e-portafolio así:

**Figura 114. Bloque e-portafolio**



Posterior a esto, se debe tener la carpeta *Uisport* dentro de la carpeta *blocks* que contiene *moodle*.

## Anexo E. PRUEBA PILOTO: Herramienta Uisport 2.0

### Instrucciones:

A continuación aparece una serie de ítems relativos a la herramienta computacional Uisport 2.0, instalada dentro del curso Ingeniería de Software I en Moodle. La herramienta busca registrar las evidencias en las etapas de análisis y diseño del proceso de desarrollo de software.

Tenga en cuenta la siguiente escala para valorar cada enunciado:

5- Acuerdo total

4- Acuerdo parcial

3- Ni en acuerdo, ni en desacuerdo

2- Desacuerdo parcial

1- Desacuerdo total

**Tabla 10. Encuesta Prueba Piloto**

FRASE	5 Acuerdo total	4 Acuerdo parcial	3 Ni acuerdo ni desacuerdo	2 Desacuerdo parcial	1 Desacuerdo total
1. Los contenidos presentados en la herramienta son suficientes y pertinentes para trabajar el proceso de desarrollo de software					
2. Utilizar el programa facilita el aprendizaje de los contenidos de la asignatura de Ingeniería de Software					
3. La herramienta permite ser autónomo en el proceso de aprendizaje					
4. Las evidencias que se pueden registrar en cada etapa del proceso de software son suficientes					
5. La herramienta apoya las diferentes etapas del					

FRASE	5 Acuerdo total	4 Acuerdo parcial	3 Ni acuerdo ni desacuerdo	2 Desacuerdo parcial	1 Desacuerdo total
proceso de desarrollo de software					
6. La herramienta permite el trabajo en equipo					
7. La herramienta permite la comunicación con el docente					
8. Las ayudas de la herramienta son consistentes y apoyan el uso de <i>Uisport</i>					
9. La interfaz es de fácil navegabilidad					
10. La herramienta apoya el aprendizaje de los contenidos del proceso de desarrollo de software					

### OBSERVACIONES DEL MATERIAL EDUCATIVO COMPUTARIZADO *UISPORT 2.0*

Que observaciones y/o recomendaciones considera usted para el mejoramiento de la herramienta computacional *Uisport 2.0*

---



---



---

**Gracias por su colaboración**

(Galvis, 1992)