

UNIVERSIDAD NACIONAL DE CAJAMARCA

FACULTAD DE INGENIERÍA

ESCUELA ACADÉMICO PROFESIONAL DE INGENIERÍA DE SISTEMAS



“IMPACTO DE LA IMPLEMENTACIÓN DE UN SISTEMA HELP DESK EN LA GESTIÓN DE INCIDENTES INFORMÁTICOS EN LA UNIDAD DE INFORMÁTICA Y SISTEMAS DE LA MUNICIPALIDAD PROVINCIAL DE CAJAMARCA”

TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

AUTOR

Bach. DAVID VÁSQUEZ COTRINA

ASESOR

Dr. Ing. MANUEL ENRIQUE MALPICA RODRÍGUEZ

Cajamarca – Perú

Septiembre 2022

COPYRIGHT © 2022
DAVID VÁSQUEZ COTRINA
Todos los derechos reservados ®

Agradecimiento

Agradezco en primer lugar a Dios, porque en su gracia me permite realizar mis objetivos. También a mis padres y hermanos porque son el soporte para mi superación. Finalmente, a todas las personas que con su apoyo hicieron posible la realización de este anhelado objetivo y en especial a mi hermanita Marisel Vásquez Cotrina por su apoyo interminable y también a mi asesor Ing. Manuel Malpica Rodríguez por su paciencia y conocimientos brindados.

Dedicatoria

Dedico esta tesis a mis padres por su esfuerzo y sacrificio que hicieron a lo largo de la vida por nuestra familia y que gracias a sus oraciones me han permitido seguir el camino de Dios. Y de manera muy especial a mi amada hermana Marisel Vásquez Cotrina, por su ejemplo de lucha, esfuerzo, mansedumbre e inagotable fe.

Contenido

<i>RESUMEN</i>	<i>xii</i>
<i>ABSTRACT</i>	<i>xiii</i>
<i>CAPÍTULO I. INTRODUCCIÓN</i>	<i>1</i>
<i>CAPÍTULO II. MARCO TEÓRICO</i>	<i>4</i>
2.1 ANTECEDENTES TEÓRICOS DE LA INVESTIGACIÓN.....	4
2.2 BASES TEÓRICAS	6
2.2.1 GESTIÓN DE INCIDENTES.....	6
2.2.2 SISTEMA HELP DESK.....	7
2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS	12
<i>CAPÍTULO III. MATERIALES Y MÉTODOS</i>	<i>13</i>
3.1 PROCEDIMIENTO	13
3.1.1 PLANIFICACIÓN	14
3.1.1.1 REQUISITOS DE ALTO NIVEL DEL PROYECTO	14
3.1.1.2 RIESGOS DEL PROYECTO.....	14
3.1.1.3 ROLES Y RESPONSABILIDADES.....	14
3.1.1.4 REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.....	15
3.1.1.5 PRESUPUESTO	15
3.1.1.6 BACKLOG DEL PRODUCTO	17
3.1.1.7 CRONOGRAMA	18
3.1.2 SPRINT 01.....	20
3.1.2.1 SPRINT BACKLOG	20
3.1.2.2 DESARROLLO DEL SPRINT	21
3.1.2.3 GRÁFICO BURDOWN	43
3.1.2.4 REVISIÓN, PRUEBAS Y RETROALIMENTACIÓN DEL SPRINT 1	44
3.1.3 SPRINT 02.....	45
3.1.3.1 SPRINT BACKLOG	45
3.1.3.2 DESARROLLO DEL SPRINT	46
3.1.3.3 GRÁFICO BURDOWN	72
3.1.3.4 REVISIÓN, PRUEBAS Y RETROALIMENTACIÓN DEL SPRINT	74

3.1.4 SPRINT 03	75
3.1.4.1 SPRINT BACKLOG	75
3.1.4.2 DESARROLLO DEL SPRINT	76
3.1.4.3 GRÁFICO BURDOWN	99
3.1.4.4 REVISIÓN, PRUEBAS Y RETROALIMENTACIÓN DEL SPRINT	100
3.2 TRATAMIENTO Y ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS.....	101
3.2.1 TRATAMIENTO	101
3.2.1.1 VARIABLES E INDICADORES.....	101
3.2.1.2 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS	102
3.2.1.3 VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS.....	102
3.2.2 ANÁLISIS DE DATOS.....	102
3.2.2.1 DIMENSIÓN TIEMPO.....	102
3.2.2.2 DIMENSIÓN SATISFACCIÓN DEL EQUIPO SOPORTE	111
3.2.2.3 DIMENSIÓN PRODUCTIVIDAD	115
3.2.3 PRESENTACIÓN DE RESULTADOS.....	118
3.2.3.1 USABILIDAD	118
3.2.3.2 TIEMPO.....	120
3.2.3.3 SATISFACCIÓN DEL EQUIPO DE SOPORTE	122
3.2.3.4 PRODUCTIVIDAD.....	123
<i>CAPÍTULO IV. ANÁLISIS Y DISCUSIÓN DE RESULTADOS.....</i>	<i>124</i>
<i>CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES.....</i>	<i>127</i>
5.1 CONCLUSIONES.....	127
5.2 RECOMENDACIONES	129
<i>REFERENCIAS BIBLIOGRÁFICAS.....</i>	<i>130</i>
<i>ANEXOS.....</i>	<i>133</i>
Anexo 1: Instrumentos para la recolección de datos.....	133
Anexo 2: Coeficiente estadístico de alfa de cronbach aplicado a los cuestionarios	137
Anexo 3: Validación de instrumentos de recolección de datos	139
Anexo 3: Pruebas de normalidad para la dimensión tiempo.....	143
Anexo 4: Tabla de General Electric.....	145
Anexo 5: Recolección de datos dimensión tiempo.	146

Anexo 6: Recolección de datos dimensión satisfacción del equipo de soporte.	151
Anexo 7: Recolección de datos dimensión productividad.	152
Anexo 8: Recolección de datos variable independiente.	153
Anexo 9: Proceso de gestión de incidentes con la aplicación del sistema helpdesk.	154
Anexo 10: Presentación del sistema help desk.	155

Índice de tablas

Tabla I. Requisitos de alto nivel del proyecto.	14
Tabla II. Roles y responsabilidades.....	14
Tabla III. Requerimientos no funcionales del sistema.	15
Tabla IV. Presupuesto: recuso humano.	15
Tabla V. Presupuesto: materiales e insumos.....	15
Tabla VI. Presupuesto: hardware y software.	16
Tabla VII. Presupuesto: servicios.	16
Tabla VIII. Presupuesto: resumen de costos.	16
Tabla IX. Backlog o pila del producto.	17
Tabla X. Cronograma de actividades.	18
Tabla XI. Backlog del sprint 1.	20
Tabla XII. Historia de usuario "Gestión de categorías, locales y nivel de prioridad".	21
Tabla XIII. Historia de usuario "buscar trabajador".	27
Tabla XIV. Historia de usuario "gestión de usuarios".	30
Tabla XV. Historia de usuario "inicio de sesión".	38
Tabla XVI. Prueba de funcionalidad Sprint 1.	44
Tabla XVII. Backlog del sprint 2.	45
Tabla XVIII. Historia de usuario "gestión de tickets".	46
Tabla XIX. Historia de usuario "gestión de mis tickets".	56
Tabla XX. Historia de usuario "perfil de usuario".	61
Tabla XXI. Historia de usuario "Dashboard".	67
Tabla XXII. Prueba de funcionalidad Sprint 2.	74
Tabla XXIII. Backlog de sprint 3.	75
Tabla XXIV. Historia de usuario "gestión de eventos".	76
Tabla XXV. Historia de usuario "estadísticas".	81
Tabla XXVI. Historia de usuario "reportes".	93
Tabla XXVII. Prueba de funcionalidad sprint 3.	100
Tabla XXVIII. Matriz de variables, dimensiones e indicadores de la investigación.....	101
Tabla XXIX. Dimensión tiempo, indicadores, tiempos y número de observaciones.....	104
Tabla XXX. Resumen de observaciones de tiempo empleado para registrar un incidente.....	105
Tabla XXXI. Resumen de observaciones de tiempo empleado para asignar un incidente	107
Tabla XXXII. Resumen de observaciones de tiempo empleado para la elaboración de reportes.	109
Tabla XXXIII. Resumen de nivel de satisfacción respecto al proceso de gestión de incidentes	111
Tabla XXXIV. Resumen de nivel de satisfacción respecto al proceso de gestión de incidentes	113
Tabla XXXV. Resumen de productividad del equipo de soporte.	115

<i>Tabla XXXVI. Dimensión tiempo - discusión de resultados.</i>	124
<i>Tabla XXXVII. Dimensión satisfacción - discusión de resultados.</i>	125
<i>Tabla XXXVIII. Dimensión productividad - discusión de resultados.</i>	125
<i>Tabla XXXIX. Dimensión usabilidad - discusión de resultados.</i>	126
<i>Tabla XL. Encuesta para medir la usabilidad del sistema.</i>	133
<i>Tabla XLI. Encuesta para medir la satisfacción de los usuarios del sistema.</i>	134
<i>Tabla XLII. Ficha de análisis documental.</i>	135
<i>Tabla XLIII. Ficha de observación.</i>	136
<i>Tabla XLIV. PRE-TEST - Datos recolectados "tiempo empleado para registrar un incidente".</i>	146
<i>Tabla XLV. POST-TEST - Datos recolectados "tiempo empleado para registrar un incidente".</i>	147
<i>Tabla XLVI. PRE-TEST - Datos recolectados "tiempo empleado para asignar un incidente".</i>	148
<i>Tabla XLVII. POST-TEST - Datos recolectados "tiempo empleado para asignar un incidente".</i>	149
<i>Tabla XLVIII. PRE-TEST - Datos recolectados "tiempo empleado para la elaboración de reportes".</i>	150
<i>Tabla XLIX. POST-TEST - Datos recolectados "tiempo empleado para la elaboración de reportes".</i>	150
<i>Tabla L. Datos pre-test de encuesta de satisfacción del equipo de soporte.</i>	151
<i>Tabla LI. Datos post-test de encuesta de satisfacción del equipo de soporte.</i>	151
<i>Tabla LII. Recolección de datos para dimensión productividad - pretest.</i>	152
<i>Tabla LIII. Recolección de datos para dimensión productividad – post-test.</i>	152
<i>Tabla LIV. Datos de variable independiente.</i>	153

Índice de figuras

Fig. 1. Ciclo de vida del desarrollo de software	9
Fig. 2. Ciclo Scrum	10
Fig. 3. Ubicación geográfica de la Municipalidad Provincial de Cajamarca.....	13
Fig. 4. Cronograma detallado.	19
Fig. 5. Diagrama de clases del proyecto.....	22
Fig. 6. Diagrama de clases de la HU01.....	23
Fig. 7. Clase "controller".....	24
Fig. 8. Vista de "nueva categoría".....	26
Fig. 9. Vista del sistema opción "listar categorías".	26
Fig. 10. Diagrama de clases de la HU03.....	27
Fig. 11. Prototipo "buscar trabajador".....	28
Fig. 12. Vista del sistema "buscador".....	30
Fig. 13. Prototipo "listar usuarios".	31
Fig. 14. Prototipo "agregar nuevo usuario".	31
Fig. 15. Diagrama de clases HU02.....	32
Fig. 16. Vista "nuevo usuario".	37
Fig. 17. Vista de "listar usuarios".	37
Fig. 18. Prototipo de "inicio de sesión".....	38
Fig. 19. Diagrama de clases "inicio de sesión".	39
Fig. 20. Vista de "inicio de sesión".....	42
Fig. 21. Backlog detallado del Sprint 1	43
Fig. 22. Gráfico Burdown Sprint 1	43
Fig. 23. Prototipo "listar tickets".	46
Fig. 24. Prototipo "agregar nuevo ticket".	47
Fig. 25. Diagrama de clases de HU06.....	47
Fig. 26. Vista de "agregar nuevo ticket".	55
Fig. 27. Vista de "listar tickets".	55
Fig. 28. Prototipo de "mis tickets".....	56
Fig. 29. Prototipo de "detalle de ticket"	57
Fig. 30. Diagrama de clases de HU07.....	57
Fig. 31. Vista de "ver mis tickets".	60
Fig. 32. Vista "detalle de ticket".	61
Fig. 33. Prototipo de "mi perfil".	62
Fig. 34. Diagrama de clases de HU11.....	62
Fig. 35. Vista de "mi perfil".....	66

Fig. 36. Prototipo de "panel de control".....	67
Fig. 37. Vista de dashboard.....	72
Fig. 38. Backlog detallado del sprint 2.	72
Fig. 39. Gráfico burdowndel Sprint 2.....	73
Fig. 40. Prototipo "calendario".....	76
Fig. 41. Diagrama de clases de "evento".....	77
Fig. 42. Vista de calendario.....	80
Fig. 43. Prototipo de vista estadísticas.....	81
Fig. 44. Diagrama de clases de la HU09.....	82
Fig. 45. Vista de "estadísticas".....	93
Fig. 46. Prototipo de vista de calendario.....	94
Fig. 47. Vista "reportes".....	98
Fig. 48. Resultado de consultar un reporte.	98
Fig. 49. Backlog detallado del sprint 3.	99
Fig. 50. Gráfico burdowndel Sprint 3.....	99
Fig. 51. Proceso inicial de Gestión de Incidentes.....	103
Fig. 52. Proceso inicial de Elaboración de reportes.....	103
Fig. 53. Prueba T-Student para tiempo de registro de un incidente.....	106
Fig. 54. Prueba T-Student para tiempo para asignar un incidente.....	108
Fig. 55. Prueba T-Student para tiempo para la elaboración de reportes.....	110
Fig. 56. Prueba t-student nivel de satisfacción respecto al proceso de gestión de incidentes.....	112
Fig. 57. Prueba t-student nivel de satisfacción respecto a los tiempos del proceso de gestión de incidentes. ...	114
Fig. 58. Prueba t-student nivel de satisfacción respecto a los tiempos del proceso de gestión de incidentes. ...	116
Fig. 59. Gráfico operabilidad o capacidad para ser usado.....	118
Fig. 60. Gráfico de grado de accesibilidad.....	119
Fig. 61. Gráfico de capacidad de aprendizaje.....	119
Fig. 62. Gráfico de tiempo empleado para registrar un incidente.....	120
Fig. 63. Gráfico de tiempo empleado para asignar un incidente.....	121
Fig. 64. Gráfico de tiempo empleado para la elaboración de reportes.....	121
Fig. 65. Gráfico de nivel de satisfacción respecto al proceso.....	122
Fig. 66. Gráfico de nivel de satisfacción respecto a los tiempos del proceso.....	122
Fig. 67. Gráfico de productividad del equipo de soporte.....	123
Fig. 68. Datos para validar instrumento encuesta aplicado a la variable independiente.....	137
Fig. 69. Coeficiente de Alfa de Cronbach de encuesta aplicado a variable independiente.....	137
Fig. 70. Datos para validar instrumento encuesta aplicado a la variable dependiente.....	138
Fig. 71. Coeficiente de Alfa de Cronbach de encuesta aplicado a variable dependiente.....	138
Fig. 72. Validación de experto – ficha de observación.....	139
Fig. 73. Validación de experto – cuestionario satisfacción.....	140

<i>Fig. 74. Validación de experto – ficha de análisis documental.....</i>	<i>141</i>
<i>Fig. 75. Validación de experto – cuestionario usabilidad.....</i>	<i>142</i>
<i>Fig. 76. Prueba de normalidad de tiempo empleado para registrar un incidente.</i>	<i>143</i>
<i>Fig. 77. Prueba de normalidad de tiempo empleado para asignar un incidente.</i>	<i>144</i>
<i>Fig. 78. Prueba de normalidad de tiempo empleado para la elaboración de reportes.....</i>	<i>144</i>
<i>Fig. 79. Tabla General Electric.....</i>	<i>145</i>
<i>Fig. 80. Nuevo proceso de gestión de incidentes.....</i>	<i>154</i>
<i>Fig. 81. Nuevo proceso de elaboración de reportes.</i>	<i>154</i>
<i>Fig. 82. Presentación del Sistema Help Desk al cliente.</i>	<i>155</i>
<i>Fig. 83. Sistema Help Desk en funcionamiento.</i>	<i>155</i>
<i>Fig. 84. Encargado de la mesa de ayuda utilizando el sistema help desk.....</i>	<i>156</i>

RESUMEN

La presente investigación tiene como finalidad determinar el impacto de la implementación de un Sistema Help Desk en el proceso de gestión de incidentes informáticos en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca, donde las actividades del proceso en mención se realizan de forma manual, actividades tales como: registro de incidentes, asignación de incidentes, elaboración de reportes, entre otros, lo que genera confusión y desorden en la atención de los casos, inoperatividad del personal, demoras en la atención y quejas por parte de los trabajadores; por otro lado, se suma la necesidad de contar con reportes estadísticos o información crucial que sirvan de apoyo para la toma de decisiones. En ese contexto, se plantea como solución la implementación de un sistema de información desarrollado bajo la metodología del modelo de desarrollo incremental y empleando el marco de trabajo Scrum, que permiten realizar un trabajo flexible a los cambios; así mismo, en cuanto al sistema, se ha desarrollado utilizando PHP como lenguaje de programación, Laravel como Framework y MYSQL como gestor de base de datos, de modo que, tras el uso del sistema desarrollado se determina una mejora en el proceso de gestión de incidentes informáticos reduciendo en un 82,62% el tiempo de registro, asignación y elaboración de reportes; también se registró un aumento del 86,5% en el nivel de satisfacción del equipo de soporte y una mejora del 24% en su nivel de productividad; determinado así un impacto en positivo en el proceso de gestión de incidentes informáticos.

Palabras clave: Sistema de información, Procesos, Gestión de incidentes, Scrum, Help Desk.

ABSTRACT

The purpose of this research is to determine the impact of the implementation of a Help Desk System in the management of computer incidents in the Informatics and Systems Unit of the Provincial Municipality of Cajamarca, where the activities of the process in question are carried out manually, activities such as: recording of incidents, assignment of incidents, preparation of reports, among others, which generates confusion and disorder in the care of cases, staff inoperability, delays in care and complaints by workers; On the other hand, there is the need to have statistical reports or crucial information that serve as support for decision making. In this context, the implementation of an information system developed under the methodology of the incremental development model and using the scrum framework is proposed as a solution, which allows flexible work to be carried out in response to changes; likewise, regarding the system, it has been developed using PHP as a programming language, Laravel as a Framework and MYSQL as a database manager, so that, after the use of the developed system, an improvement in the management process of computer incidents reducing by 82.62% the time of registration, assignment and preparation of reports; there was also an 86.5% increase in the satisfaction level of the support team and a 24% improvement in their productivity level; thus determining a positive impact on the computer incident management process.

Key Words: Information system, Process, Incident management, Scrum, Help Desk.

CAPÍTULO I. INTRODUCCIÓN

Las tecnologías de información han revolucionado la forma tradicional de funcionamiento de una empresa u organización, su uso se ha convertido en una necesidad. Tal es así que las empresas están implementando algún tipo de tecnología en el desarrollo de sus actividades [1]; sin embargo, estos cambios conllevan a asumir el reto de asegurar el buen funcionamiento de su infraestructura tecnológica, ya que esto se puede traducir en pérdidas o ganancias para las empresas [2]; un estudio realizado por la consultora Beservices refleja que un 33,7% de los empleados en España considera que los problemas informáticos son uno de los mayores motivos de estrés en el trabajo y ello supone un grave impacto en la productividad de la empresa [3]; dado que un trabajador puede llegar a perder entre 30 y 120 minutos de su tiempo laboral por fallos informáticos, lo que significa un promedio de 100 horas al año [4].

Por otro lado, según el estudio realizado en los Estados Unidos por EMC revela que cada año se pierde alrededor de 1,7 billones de dólares por fallos informáticos y además se desperdicia un total de 25 horas de trabajo [5]; de modo similar, las empresas en México pueden llegar a perder hasta 15 millones de dólares al año como consecuencia de fallos e interrupciones tecnológicas, lo que a su vez afectan a su reputación y productividad [6]; de forma semejante, en el Perú, según Delgado agente de Lenovo Perú, un trabajador alcanza a perder hasta un 46% de su tiempo laboral debido a fallas en los equipos informáticos, lo que repercute o afecta a la productividad tanto del trabajador como de la empresa [7].

Como vemos, las empresas en el intento de mejorar su infraestructura de TI¹ para apoyar el desarrollo de sus actividades no están libres de sufrir alguna falla en su equipamiento tecnológico y ello representa una interrupción del servicio y afectando su productividad. La Municipalidad Provincial Cajamarca no es ajena a la implementación de tecnología en el desarrollo de sus actividades, por lo que, a través de su área encargada, la Unidad de Informática y Sistemas, viene sumando diferentes soluciones tecnológicas, tanto hardware como software, para apoyar el desarrollo de sus actividades laborales y brindar un mejor servicio a los usuarios internos y a los ciudadanos; es así que, durante las actividades laborales diarias, se presentan diferentes incidentes informáticos con los cuales el equipo de encargado del soporte informático

¹ TI: Tecnologías de información.

tiene que enfrentar, sin embargo, en la actualidad la UIS² no tiene ningún tipo de sistema para apoyar a este proceso.

En estas condiciones, el registro de los incidentes informáticos se realiza de manera manual haciendo uso de una pizarra y/o hojas de papel, lo que muchas veces es extraviado generando confusión y desorden en la atención de los casos, lo que a su vez ocasiona quejas por parte de los usuarios, quienes muestran su malestar; del mismo modo, la asignación de los incidentes recibidos hacia el equipo de soporte para su atención es bastante desorganizado, ya que no existe una forma adecuada de llevar un control objetivo, lo que genera desigualdad en la asignación de casos, inoperatividad del personal y demoras en la atención. Además, a ello se suma la necesidad de contar con reportes estadísticos o información crucial que sirvan de apoyo para la toma de decisiones, por ejemplo, tiempo de atención, cantidad de atenciones o tickets, áreas que presentan más incidencias (área crítica), etc.

Dado el contexto, en el presente proyecto de investigación se formula la siguiente interrogante ¿Cuál es el impacto de la implementación de un Sistema Help Desk en la gestión de incidentes informáticos en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca? Y se plantea como hipótesis: La implementación de un Sistema Help Desk mejora la gestión de incidentes informáticos en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca.

En ese sentido, esta investigación se **justifica** de manera práctica dado que contribuye a resolver una necesidad real existente en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca, la cual es mejorar la gestión de incidencias informáticas, también se justifica de manera académica, debido a que la presente investigación va a dar a conocer cómo funciona un Sistema Help Desk, y ser referente de consulta para futuras investigaciones. Asimismo, este proyecto atiende al sentido de urgencia de la institución de contar con una herramienta exclusiva para apoyar el proceso de gestión de incidentes informáticos ya que actualmente carece de herramienta alguna. Finalmente, esta investigación presenta relevancia social, debido a que este proyecto de tesis busca brindar información útil y ser replicado en otras instituciones que enfrenten una realidad problemática similar.

En cuanto al **alcance**, el proyecto abarca el desarrollo de un sistema web como herramienta de apoyo para mejorar la gestión de incidentes informáticos, usando PHP como lenguaje de

² UIS: Unidad de Informática y Sistemas.

programación, Laravel como Framework y MYSQL como gestor de base de datos, todo ello conforme a las políticas de trabajo de la institución. Asimismo, se tiene por **objetivo principal** de la investigación: Determinar el impacto de la implementación de un Sistema Help Desk en la gestión de incidentes informáticos en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca. Y como **objetivos específicos**: primero, identificar el impacto de la implementación de un Sistema Help Desk en el tiempo empleado para registrar y asignar un incidente y para elaborar reportes en la Unidad de informática y Sistemas de la MPC³; segundo, conocer el impacto de la implementación de un Sistema Help Desk en la satisfacción del equipo de soporte de la Unidad de informática y Sistemas de la MPC y tercero, conocer el impacto de la implementación de un Sistema Help Desk en la productividad en la Unidad de informática y Sistemas de la MPC.

Finalmente, para la **organización** de este informe, se ha dividido en cinco capítulos, los cuales se detallan a continuación: el capítulo I corresponde a la introducción, donde se describe el contexto, el problema, la justificación o importancia de la investigación, los alcances de la investigación y los objetivos. El capítulo II, contiene el marco teórico, antecedentes históricos, bases teóricas y definición de términos básicos. El capítulo III aborda los materiales y métodos utilizados durante la investigación, también se especifica el tratamiento y análisis de los datos y presentación de resultados. En el capítulo IV, se presenta el análisis y discusión de resultados, donde se explica y discute los resultados de la investigación. Finalmente, en el capítulo V, presenta las conclusiones y recomendaciones del proyecto, es decir, los resultados finales obtenidos frente a los objetivos planteados, y también se indica algunas recomendaciones como aporte de este trabajo de investigación.

³ MPC: Municipalidad Provincial de Cajamarca

CAPÍTULO II. MARCO TEÓRICO

2.1 ANTECEDENTES TEÓRICOS DE LA INVESTIGACIÓN

Diversos estudios han sido desarrollados en relación al tema de la presente investigación, los cuales sirven como antecedentes. Por ejemplo, en el ámbito **internacional**, específicamente en Ecuador, los autores Ponce y Samaniego [8], en su tesis: “Análisis del impacto del help desk en los procesos de soporte técnico en una organización”, tienen por objetivo optimizar la utilización de recursos, mejorar el desempeño laboral y mejorar los tiempos de respuesta del proceso de atención del área de soporte técnico de una organización. Ante lo cual propone, adquirir e implementar un Sistema Help Desk que se adecue a sus necesidades. Luego de la implementación, obtiene como resultados que los tiempos de respuesta mejoran en un 40%. Esta investigación se considera importante ya que se enfoca en optimar un proceso ya definido y hace una comparación de diferentes sistemas help desk, mostrando requisitos y funcionalidades que debe cumplir un buen sistema de gestión de incidentes.

Por su parte, también en Ecuador, López [9], en su proyecto de investigación: “Implementación de un sistema de mesa de ayuda informático (help desk) para el control de incidencias que se presentan en el Gobierno Autónomo Descentralizado de la provincia de Esmeraldas”, tiene por objetivo optimizar el proceso de asistencia técnica informática existente en dicha institución. Ante lo cual que propone implementar un software help desk, y de esa forma contribuir además con el desarrollo tecnológico de la entidad. Como resultados, se aprecia que la institución ha mejorado notablemente su proceso de gestión de incidencias, ya que con el uso del sistema existe un incremento del 99% en la capacidad de atención. Esta tesis se considera de ayuda ya que hace uso de técnicas e instrumentos de recolección de datos similares a la presente investigación.

En el ámbito **nacional**, Hoyos [10], en su tesis “Sistema Informático Help Desk vía Web y Móvil para mejorar el control de incidencias en la Unidad de Tecnologías de Información de la Municipalidad Distrital de Pacasmayo” tiene como objetivo mejorar el control de incidencias, desde su reporte hasta su finalización, para lo cual propone implementar un sistema help desk tanto web cómo móvil utilizando el lenguaje de programación PHP. Es así que luego de la implementación, obtiene como resultados que el tiempo de registro de un incidente se ha reducido en 80,65%, el tiempo de búsqueda de casos relacionados con la incidencia en 87.25% y un incremento en 22% en el nivel de satisfacción del personal. Esta tesis se considera de

referencia debido a que analiza y desarrolla un sistema web utilizando el lenguaje de programación PHP; herramientas propuestas similares a la presente investigación.

Del mismo modo, Cruz [11], en su tesis “Sistema help desk para mejorar el control de incidencias de hardware y software bajo la modalidad open source en la Gerencia Regional de Salud La Libertad” tiene como objetivo optimizar el proceso de control de eventos de tipo informático en dicha institución, ante lo cual propone la implementación de un sistema Help Desk como medida para alcanzar tal objetivo. Obtiene como resultados que la implementación del sistema mejora en un 56% el tiempo de registro de un incidente, en un 49% el tiempo empleado en la resolución de un evento, y disminuye la acumulación de incidencia reportadas en un 56%. Esta tesis se considera importante ya que utiliza el mismo diseño de investigación (pre - experimental) que el presente proyecto de tesis.

Y en el ámbito **local**, Cajamarca, Los investigadores Chavarry y Gallardo [12], en su tesis: “Influencia de un sistema de help desk en la gestión de incidencias de tecnologías de información, de la Municipalidad Distrital de Llacanora periodo - 2017”, tienen por objetivo definir una forma sistematizada y estandarizada para el proceso de atención de incidentes que brinda el área de soporte técnico de la Municipalidad en mención. Es así, que proponen implementar un sistema help desk desarrollado acorde a las necesidades. Logrando, una mejora tanto en la forma de gestionar estos incidentes, como en la perspectiva que tienen los usuarios hacia el servicio recibido. Así como también, un incremento en el nivel de satisfacción de 1.29 a 3.39. Esta tesis resulta beneficiosa debido a que el estudio se realiza en una institución pública de la misma región al igual que el presente proyecto de tesis.

Finalmente, Ibáñez [13], en su tesis “Impacto de la implementación de gestión de incidentes de TI del Framework ITIL V3 en la Sub-Área de End User Computer en Goldfields La Cima S.A. – Operación minera Cerro Corona”, tiene por objetivo optimizar la calidad del servicio de soporte que brinda el área de TI. Para lo cual, propone establecer mejoras en el proceso, como utilizar el marco de trabajo ITIL e implementar un software help desk que ayude en la gestión de incidentes. Los resultados obtenidos muestran incremento en la capacidad de atención de incidentes en 113%, reducción del tiempo de atención de un incidente, desde su apertura hasta su cierre, en un 50% y una mejora del 70% en el nivel de satisfacción de los usuarios. Esta investigación es importante ya que enseña cómo manejar un proceso de gestión de incidentes usando como referencia el marco de trabajo ITIL.

2.2 BASES TEÓRICAS

2.2.1 GESTIÓN DE INCIDENTES

Se entiende por incidente a una interrupción no proyectada que se da en un servicio de Tecnología de Información (TI) al cual representa una reducción en la calidad de este servicio. También, el fallo de un componente que aún no haya afectado al servicio también de TI se toma como una incidencia. Por lo tanto, la gestión de incidencias, es el proceso que cubre todo tipo de incidencias, ya sean fallos, preguntas o consultas reportadas por usuarios, personal técnico o bien descubiertas automáticamente por diversas herramientas disponibles. Dicho de otro modo, es el cómo se administra un incidente o requerimiento, que va desde su registro hasta su cierre o finalización [14].

Su **objetivo principal** es restaurar cuanto antes el funcionamiento habitual del servicio afectado, tratando en lo posible de reducir el impacto negativo en las operaciones de negocio [14]. Pero también se tiene otros objetivos tales como: detectar cualquier alteración en los servicios de TI, registrar dichas alteraciones y clasificarlas, asignar el personal encargado de restaurar el servicio, mejorar la productividad de los usuarios, mayor monitorización y control de los procesos del servicio, optimización de recursos disponibles y mejora la satisfacción general de los usuarios y clientes [15].

Además, dentro de sus **fases** o etapas tenemos: Identificación: el incidente es detectado o reportado. Registro: el incidente es creado en el registro. Categorización: el incidente es codificado por tipo, estado, urgencia, impacto, etc. Priorización: es el cómo el incidente va ser manejado por el personal de soporte. Escalada: el incidente va cambiado de personal asignado hasta encontrar solución. Investigación y diagnóstico: si no se conoce la solución para el incidente, se debe investigar. Resolución: Cuando se detecta una solución potencial, debe aplicarse y probarse. El incidente se considera resuelto. Cierre: Antes de cerrar el problema, el equipo de help desk debe validar que el usuario está satisfecho con la resolución del problema, que el cierre ha sido categorizado, que los datos estén completos y verificar si se trata de un problema recurrente [16].

Una correcta gestión de las incidencias conlleva a la obtención de **beneficios** tales como: mayor control en la monitorización del servicio, aumento de la productividad de los usuarios al disminuir los tiempos de indisponibilidad de los servicios, optimización de los recursos destinados al soporte del servicio, y adquirir una base de conocimiento amplia y bien

estructurada que permite identificar rápidamente las acciones a llevar a cabo a la hora de restaurar un servicio [17].

Además, la gestión de incidentes de TI tiene algunos **indicadores** clave que miden el rendimiento de dicho proceso, con la finalidad de realizar una gestión eficaz. Según ITIL propone los siguientes: tiempo para resolver un incidente, tiempo de respuesta, tasa de satisfacción del usuario final, entre otros [18]. Por ello, el presente proyecto se consideró el tiempo (asignar un incidente, elaboración de reportes, registro de incidente), satisfacción del equipo de soporte (nivel de satisfacción con respecto al proceso y al tiempo) y productividad (Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo).

2.2.2 SISTEMA HELP DESK

En primer lugar, se tiene al **help desk** que traducido al español es mesa de ayuda, viene a ser el conjunto de servicios que permiten gestionar incidencias, requerimientos y atenciones relacionadas con las tecnológicas de información y brindar una solución para asegurar su buen funcionamiento [19]. De manera similar, Tenorio [20], define help desk como un conjunto de recursos tanto tecnológicos como humanos, que trabajan de forma interrelacionada para gestionar y solucionar todas las posibles incidencias de manera integral, junto con la atención de requerimientos relacionados a las Tecnologías de la Información y la Comunicación (TIC)”. Dentro de sus funciones principales se tiene [21]:

- Recepcionar incidentes: Help desk es el encargado de recepcionar todas las notificaciones de los usuarios utilizando uno o más canales tales como: teléfono, fax, portal web o correo electrónico.
- Llevar un registro de incidencias: los incidentes que son reportados deben ser documentados: tipo, involucrados, sistemas, dispositivos y servicios afectados.
- Priorizar incidentes: el personal encargado debe ser muy calificado y estar presto a responder de forma rápida según la priorización del incidente.
- Clasificación de incidentes: dependiendo de los datos registrados del incidente, se clasifica y se procede a su atención inmediata o a la cola de espera.
- Seguir las recomendaciones y procesos: a parte de la solución brindada se debe tener en cuenta la información inherente de los equipos o softwares.
- Escalar todas las incidencias según sea necesario: La escalada es importante porque asegura el cumplimiento del acuerdo de servicio (SLA).

- Reunir los recursos necesarios: el help desk puede tener otras funciones según el acuerdo de servicio establecido.
- Mantener una comunicación constante con todas las partes: Se debe mantener una comunicación constante con las partes involucradas tales como: usuarios finales, administradores, recursos en el sitio y proveedores de terceros.
- Realizar todas las actividades programadas: esto depende del acuerdo de servicio establecidos por la institución.

Luego, se plantea la definición de lo que es un **sistema informático**, el cual se define como una colección de elementos (hardware, software y usuario) que están diseñadas para recibir, procesar, administrar y presentar información en un formato significativo. Este se compone de la parte física o hardware, como, por ejemplo, computadoras u otro dispositivo. También está la parte lógica o software o también conocidos como aplicaciones o programas. Y finalmente tenemos al usuario, que es la persona que maneja los programas y periféricos [22].

Por lo tanto, para la presente investigación, un **sistema help desk** es un software que almacena información de los clientes y de las incidencias que estos reportan, generándolos en forma de tickets para luego asignarlos o derivarlos con una persona encargada de su atención, todo ello para un adecuado control, que va desde el registro hasta su cierre [23].

En ese sentido, es imprescindible hacer referencia al **desarrollo de software** dado que el presente proyecto abarca el desarrollo de un sistema. Por lo tanto, el desarrollo de software es el conjunto de actividades dedicadas al proceso de creación, diseño, implementación y soporte de software. El proceso de desarrollo de software contiene fases tales como: análisis de requerimientos, que es donde se reúne todos los requerimientos, características, funcionalidades del cliente con respecto al software a desarrollar; diseño, aquí es donde se ve el tema del lenguaje de programación a utilizar, el tipo de base de datos, etc. es decir se define su arquitectura; desarrollo, es donde empieza el proceso de construcción del software; pruebas, es donde se examina los resultados de la fase anterior; Implementación, esta fase se refiere a la puesta en uso del software y finalmente, la fase de mantenimiento, una vez concluidos las fases anteriores y levantadas las posibles observaciones, el software pase a esta última etapa que abarca actualizaciones ante posibles cambios o mejoras; estas fases se observan en la siguiente figura [24] que muestra el ciclo de vida del desarrollo de software:

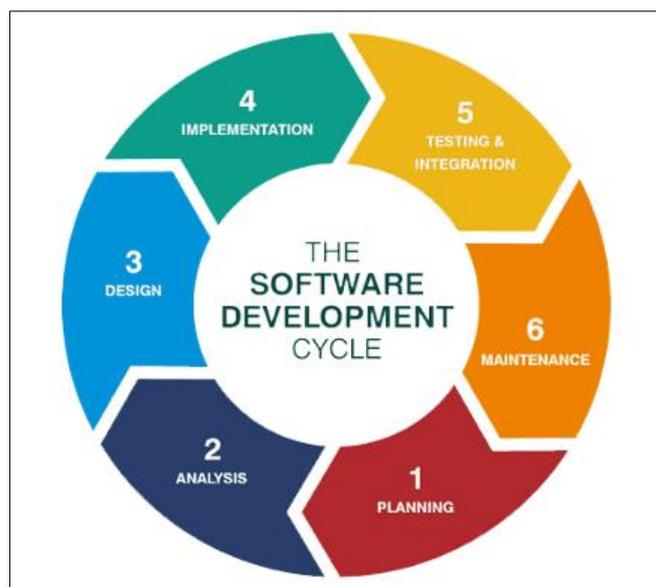


Fig. 1. Ciclo de vida del desarrollo de software

Además, el desarrollo de software expone una serie de **modelos y metodologías**. Dentro de los más importantes tenemos: **Modelo en Cascada**, el cual es el más antiguo en el desarrollo de software, se le atribuye este nombre debido al ordenamiento de sus fases; en este modelo no podemos saltar los pasos, sino que debemos apegarnos rigurosamente a ellos, no se puede avanzar sin antes terminar y revisar la fase anterior, ello genera una desventaja ya que, si un cliente cambia de opinión en algún aspecto del software, en teoría no se podría volver atrás y eso significaría retrasos en el tiempo de entrega; además, para ver los resultados o alguna funcionalidad del sistema, se tiene que esperar hasta que el proyecto se encuentre muy avanzado o, dicho de otra manera, terminado [25].

También está el **Modelo de Desarrollo Incremental**, a diferencia del modelo en cascada, permite ir desarrollando el sistema de forma iterativa, donde en cada etapa va agregando alguna funcionalidad; además, los tiempos de entrega son mucho más cortos permitiendo realizar pruebas constantes, lo que resulta en realizar cambios de manera sencilla; algo importante también es que los clientes pueden aclarar los requisitos planteados al inicio, ya que quizás no lo tenían bien analizado; además, el cliente no tiene que esperar hasta el final para ir usando el software, sino que en cada entrega puede ir probando su funcionalidad, pero si se tiene que poner mucho énfasis en la planeación [25].

Sin embargo, ante los inconvenientes que presentaban los modelos tradicionales de desarrollo de software, surge una metodología enfocada en la flexibilidad y la rapidez. **Las metodologías ágiles**, las cuales son una forma de manejar o gestionar un proyecto dividiéndolo en varias fases

o en pequeñas partes, buscando conseguir una colaboración constante con los interesados y una mejora continua en cada etapa. Con el objetivo de desarrollar productos y servicios de calidad que cumplan las necesidades de los clientes [26]. Dentro de sus principales ventajas tenemos [27]: implementación rápida de soluciones, mayor flexibilidad y adaptabilidad al cambio, mejor utilización de recursos, mayor calidad, tiempos de entrega rápidos, detección más rápida de problemas y defectos en el proceso, mayor atención a las necesidades de los clientes y mayor colaboración entre los involucrados.

Dentro de estas metodologías existen varias opciones, tales como: Scrum, Programación Extrema (XP) y Kanban. Aunque **Scrum**, es considerada un marco de trabajo más que una metodología, es la más utilizada, ya que es muy flexible y adaptable a cualquier proyecto, no tiene pasos exigentes, sino que podemos extraer algunos y adaptarlos en nuestro proyecto, por ello es utilizada por muchos equipos de trabajo para llevar a cabo proyectos como el desarrollo de software. El fundamento de Scrum son los ciclos cortos o iteraciones (sprint), los cuales están conformados por una serie de tareas que producen un resultado concreto, tal como se presenta en la figura siguiente exponiendo el ciclo de vida de Scrum [28]:

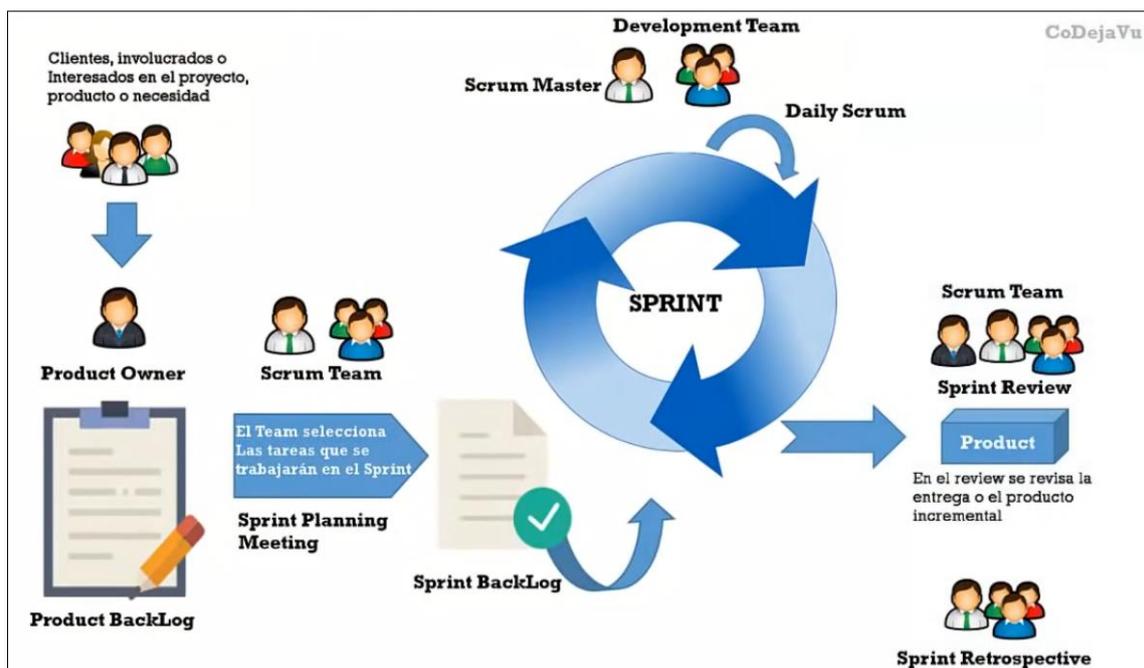


Fig. 2. Ciclo Scrum

Para llevar a cabo un proyecto, Scrum recomienda las siguientes **fases** [28]: Planificación: Crear visión del proyecto, identificar roles Scrum Master y Stakeholders, formar el equipo Scrum, desarrollar épicas o historias de usuarios, crear backlog y planificar lanzamiento. Ejecución y Estimación: Crear historias de usuarios, estimar historias de usuarios, identificar tareas, estimar

tareas y crear backlog sprint. Implementación: Crear entregables, realizar daily standup o reuniones y refinar el backlog. Revisión y Retrospectiva: Demostrar y validar el sprint y Lanzamiento: Enviar entregables y retrospectiva del proyecto.

Del mismo modo, dentro Scrum cada involucrado tiene un **rol**, por ejemplo [29]: Dueño del producto (Product Owner): es el contacto entre el equipo que desarrolla el proyecto y el cliente. Por lo que en este perfil se requiere tener mucho conocimiento del negocio para priorizar las tareas y que el equipo de desarrollo entregue algo funcional en poco tiempo. Scrum Master: es un experto en Scrum, y su función es que el equipo de desarrollo trabaje según las guías de Scrum, no es un jefe, sino más bien, es un facilitador para el equipo, además está a cargo de las ceremonias de Scrum, como las reuniones diarias (daily meeting). Equipo de desarrollo (Development Team): es equipo de profesionales que se encargan de crear el equipo, son personas con mucha capacidad técnica.

Por otra parte, es importante indicar que todo software o sistema debe cumplir con ciertos requisitos de calidad, en tal sentido la ISO (Organización de Estándares Internacionales) dispone de la familia ISO 25000, para especificar y medir características de calidad de cualquier producto de software [30]. Por ejemplo, una de las dimensiones más comunes a emplear es la usabilidad, que viene a ser la capacidad del software para ser entendido, aprendido, usado y resultar atractivo para el usuario, dentro del cual se tiene como **indicadores** de medición al grado de operabilidad, grado de capacidad de aprendizaje y grado de accesibilidad [31].

Para la presente investigación se optó por emplear el modelo incremental, juntamente con Scrum para el desarrollo del software o sistema help desk, así como también emplear **UML** (Lenguaje Unificado de Modelado) como la notación para la documentación ya que ofrece una serie de modelos, como los diagramas de clases, objetos, casos de uso, estados, secuencias, componentes, etc. [32]. Esto es importante porque sirve para una mejor interpretación y comprensión del sistema tanto para el equipo de desarrollo como para los clientes [33].

2.3 DEFINICIÓN DE TÉRMINOS BÁSICOS

SLA: Por sus siglas en inglés “Service Level Agreement” o Acuerdo de nivel servicio. Se trata de un acuerdo o contrato firmado donde se establecen deberes y responsabilidades entre un proveedor de servicios y el cliente [21].

ITIL: Las siglas ITIL significan “Information Technology Infrastructure Library”, que traducido literalmente es Biblioteca de Infraestructura de Tecnologías de Información. Es decir, es una guía de buenas prácticas para la gestión de servicios de tecnologías de la información (TI) [2].

TIC: Viene de las siglas Tecnologías de Información y Comunicación, que viene a ser el uso de cualquier computador, dispositivo de almacenamiento, red, infraestructura y proceso para crear, procesar, almacenar, proteger e intercambiar todas las formas de datos electrónicos [34].

UIS: Son las siglas de la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca.

Reportante: Es la persona que se comunica con la mesa de ayuda y reportar un incidente del tipo informático y solicita apoyo para que un personal del equipo de soporte pueda atender dicho caso [23].

Incidente Informático: Es todo evento que represente un inconveniente en el normal funcionamiento de cualquier equipo informático, tanto en hardware o software, y que ello dificulte o detenga el desarrollo de sus labores en la institución [14].

Ticket: Se considera a la solicitud que ingresa al Sistema Help Desk y el encargado lo registra creando un ticket para atender el caso particular de ese usuario y que contiene toda la información de la interacción entre help desk y reportante [23].

Asignar ticket: Es el proceso donde el encargado de la mesa de ayuda canaliza el incidente registrado hacia un personal del equipo de soporte a través del Sistema Help Desk [23].

CAPÍTULO III. MATERIALES Y MÉTODOS

La investigación se ha llevado a cabo en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca, la cual está ubicada en la región Cajamarca en la Av. Alameda de los Incas. Asimismo, el desarrollo de la investigación se proyectó entre los meses de abril a noviembre del año 2021, sin embargo, debido a la coyuntura dado por la pandemia del covid-19, los plazos se han alargado. A continuación, en la fig.3 se muestra un mapa con la ubicación geográfica de la institución en mención.

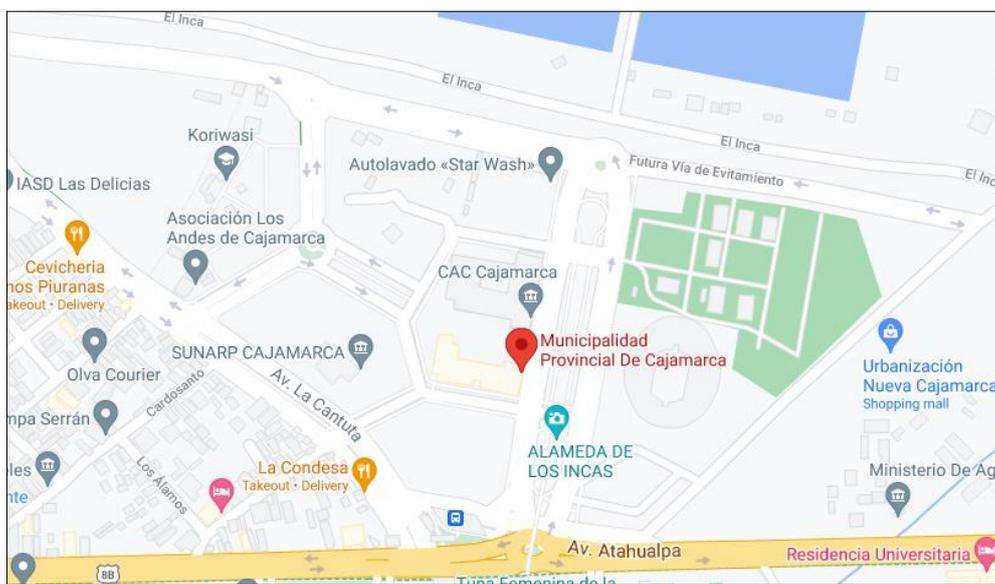


Fig. 3. Ubicación geográfica de la Municipalidad Provincial de Cajamarca

3.1 PROCEDIMIENTO

Para el desarrollo de la presente investigación se empleó el marco de trabajo Scrum juntamente con el modelo de desarrollo incremental. Se adopta estas herramientas ya que el modelo incremental permite ir añadiendo funcionalidades durante el proceso del desarrollo del software, y Scrum, dado que se adapta al proyecto, es muy flexible a los cambios que el cliente pueda tener referente al sistema, lo cual permite realizar un trabajo más dinámico. Además, se emplea también la notación UML para realizar la especificación y documentación del software a desarrollar. Por lo tanto, bajo este enfoque, se inició con la elaboración de los requerimientos del sistema o pila del producto, que luego se dividió en ciclos cortos o iteraciones denominados Sprint, el cual una vez finalizado da como resultado un entregable, que viene a ser un incremento en el desarrollo del proyecto total.

3.1.1 PLANIFICACIÓN

Durante esta fase se estableció los lineamientos del proyecto, como, los requisitos de alto nivel del proyecto, los riesgos que pueden afectar al proyecto; también se identificó a los involucrados (roles y responsabilidades). Además, se desarrolló las historias de usuarios, el cual contiene los requerimientos del cliente para el sistema, con lo cual se creó el backlog o pila del producto, presupuesto, cronograma y finalmente se planificó los sprint en base a las prioridades.

3.1.1.1 REQUISITOS DE ALTO NIVEL DEL PROYECTO

Tabla I. Requisitos de alto nivel del proyecto.

Requisitos	Criterio de Éxito
Acceso a las bases de datos de la Municipalidad Provincial de Cajamarca.	Apoyo del administrador de base de datos.
Disponibilidad de los interesados.	Reuniones para una adecuada definición de requerimientos del sistema.
Respetar el cronograma	Finalizar el proyecto a tiempo

3.1.1.2 RIESGOS DEL PROYECTO

- Demora en la obtención de información de las Bases de Datos.
- Fallas en los servidores del Data Center de la institución.
- Indisponibilidad del equipo Scrum debido a situaciones que afecten la salud de los involucrados.
- Retrasos en la ejecución del proyecto.

3.1.1.3 ROLES Y RESPONSABILIDADES

Tabla II. Roles y responsabilidades

ROL	RESPONSABLES
Dueño del producto (Product Owner)	Ing. Wilson Becerra Pérez
Scrum Master	Bach. David Vásquez Cotrina
Equipo Scrum (Scrum Team)	Ing. Jonny Cálua Tasilla Bach. David Vásquez Cotrina

3.1.1.4 REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA

Tabla III. Requerimientos no funcionales del sistema.

Requerimientos no funcionales
<ul style="list-style-type: none">➤ El sistema debe desarrollarse con el framework Laravel y PHP➤ Se debe utilizar MYSQL como gestor de base de datos.➤ Solo los trabajadores activos (contrato vigente) deben poder gestionar un ticket de atención.➤ La creación de usuarios dentro del sistema debe ser previa solicitud al jefe de área.➤ El sistema de ser web.➤ El sistema debe ser accesible desde una computadora y celular, sin distorsionar su apariencia.➤ El sistema debe ser fácil de usar, sin complicaciones.➤ El sistema debe mantener textos legibles y entendibles.➤ El sistema debe contener el logo de la actual gestión municipal.

3.1.1.5 PRESUPUESTO

El presupuesto para el desarrollo del proyecto será asumido en su totalidad por el tesista y la institución. La Municipalidad dispone de equipamiento y licencias de softwares necesarios para el desarrollo de este proyecto, por lo que dichos costos no se han considerado.

RECURSO HUMANO:

Tabla IV. Presupuesto: recuso humano.

PERSONAL	FUNCION
Bach. Vásquez Cotrina David	Tesista
Ing. Malpica Rodríguez Manuel Enrique	Asesor

MATERIALES E INSUMOS:

Tabla V. Presupuesto: materiales e insumos.

MATERIAL	UNIDAD DE MEDIDA	CANTIDAD	PRECIO UNITARIO (S/.)	TOTAL (S/.)
Lapiceros	Unidad	10	0.50	5.00
Papel bond A4 80g	Millar	06	30.00	180.00
Folder manila	Unidad	20	0.50	10.00
USB 16 GB	Unidad	01	40.00	40.00
			TOTAL (S/.)	235.00

HARDWARE Y SOFTWARE:

Tabla VI. Presupuesto: hardware y software.

CLASIFICACIÓN	RECURSO	SUB TOTAL
HARDWARE	Laptop	S/. 2500.00
	Servidor	-
	Impresora	S/. 600.00
SOFTWARE	MYSQL	-
	Visual Studio Code	-
	Microsoft Office	-
	Navicat	-
	TOTAL	S/.3100.00

SERVICIOS:

Tabla VII. Presupuesto: servicios.

SERVICIOS	UNIDAD MEDIDA	CANTIDAD	PRECIO UNITARIO (S/.)	TOTAL (S/.)
Anillados	Unidad	10	5.00	50.00
Internet	Horas	350	1.00	350.00
Empastado	Unidad	06	30.00	180.00
Consume eléctrico	Mes (Kw/hora)	05	40.00	200.00
Fotocopias	Unidad	100	0.10	10.00
			TOTAL (S/.)	690.00

RESUMEN DE COSTO:

Tabla VIII. Presupuesto: resumen de costos.

CONCEPTO	SUBTOTAL (S/.)
Materiales e insumo	235.00
Servicios	690.00
Hardware y software	3100.00
TOTAL (S/.)	4025.00

3.1.1.6 BACKLOG DEL PRODUCTO

Así también, se elaboró el Backlog o pila del producto, donde se muestra la lista de requerimientos por parte del cliente, dichos requerimientos se formularon en forma de historias de usuarios. Así mismo, se ordenó la pila en base al valor, riesgos y necesidades del negocio, en ese contexto, se dividió el proyecto en 03 Sprints o iteraciones. A continuación, se detalla la pila del producto juntamente con los actores que establecieron las historias de usuarios.

Los actores o roles que se ha considerado para realizar las historias de usuarios son:

- ✓ **Supervisor:** es la persona que está a cargo de todo el equipo de soporte informático de la UIS.
- ✓ **Encargado de la mesa de ayuda:** es la persona responsable de atender los diferentes canales de atención y de gestionar los incidentes informáticos reportados.
- ✓ **Personal de soporte:** es la persona encargada de atender los casos asignados por el encargado de la mesa de ayuda.

A continuación, se muestra la tabla detallada del backlog del producto, el cual contiene las historias de usuario con sus prioridades.

Tabla IX. Backlog o pila del producto.

ID	Como (Rol)	Deseo ...	Prioridad	Sprint	Estimación
HU01	Supervisor	Gestionar (registrar, listar, editar, eliminar) categorías de incidente, locales municipales y niveles de prioridad de incidentes.	1	1	7
HU03	Encargado de la mesa de ayuda	Buscar trabajadores y áreas. De acuerdo a su DNI, nombres y apellidos.	2	1	5
HU02	Supervisor	Crear, editar, eliminar o bloquear un usuario del sistema, además poder asignarle permisos de acceso (rol) dentro del sistema.	3	1	3
HU08	Supervisor	Iniciar sesión en el sistema	4	1	3
HU06	Encargado de la mesa de ayuda	Gestionar (registrar, editar, eliminar, listar) un incidente y asignarle un personal responsable de su atención	5	2	5
HU07	Equipo de soporte	Gestionar los tickets a mi cargo (listar, cambiar de estado, agregar un comentario).	6	2	4
HU11	Equipo de soporte	Tener un apartado donde poder administrar mi perfil, actualizar mis datos, subir foto y cambiar mi contraseña.	7	2	3

ID	Como (Rol)	Deseo ...	Prioridad	Sprint	Estimación
HU04	Encargado de la mesa de ayuda	Tener un dashboard personalizado según el usuario.	8	2	3
HU05	Supervisor	Tener un calendario y poder agregar, editar y eliminar un evento en una fecha seleccionada.	9	3	3
HU09	Supervisor	Tener un apartado donde se muestre estadísticas acerca de la atención de incidentes y la productividad del equipo de soporte. Por ejemplo: cantidad de tickets según la categoría del incidente, áreas u oficinas que solicitan más atenciones, y cantidad de tickets registrados por mes.	10	3	7
HU10	Encargado de la mesa de ayuda	Generar el reporte en Excel con las atenciones de los incidentes registrados.	11	3	4

3.1.1.7 CRONOGRAMA

En la siguiente tabla se presenta el cronograma de actividades a realizar en el desarrollo del presente trabajo de investigación.

Tabla X. Cronograma de actividades.

Actividad	Inicio	Fin	Duración
Definición y planeamiento del proyecto.	10/05/2020	11/07/2021	25 días
Definición del alcance del proyecto.			
Identificar y definir roles y stakeholders.	14/06/2021	09/07/2021	20 días
Crear el backlog del producto.	12/07/2021	06/08/21	20 días
Crear y estimar historias de usuarios.			
Definir y planificar Sprints.	09/08/2021	03/09/2021	20 días
Desarrollo del Sprint 1			
Pruebas o testing del Sprint 1			
Reunión de revisión y retrospectiva del sprint 1.	06/09/2021	01/10/2021	20 días
Desarrollo del Sprint 2			
Pruebas o testing del Sprint 2			
Reunión de revisión y retrospectiva del sprint 2	04/10/2021	29/10/2021	20 días
Desarrollo del Sprint 3			
Pruebas o testing del Sprint 3			
Reunión de revisión y retrospectiva del sprint 3	01/11/2021	12/11/2021	10 días
Envío de entregables			
Puesta en producción del sistema	15/11/2021	17/12/2021	30 días
Tratamiento y análisis de datos			
Elaboración de informe de los resultados	20/12/2021	07/01/2022	15 días

La siguiente figura (fig.4) muestra el gráfico del cronograma de actividades de la tabla anterior.

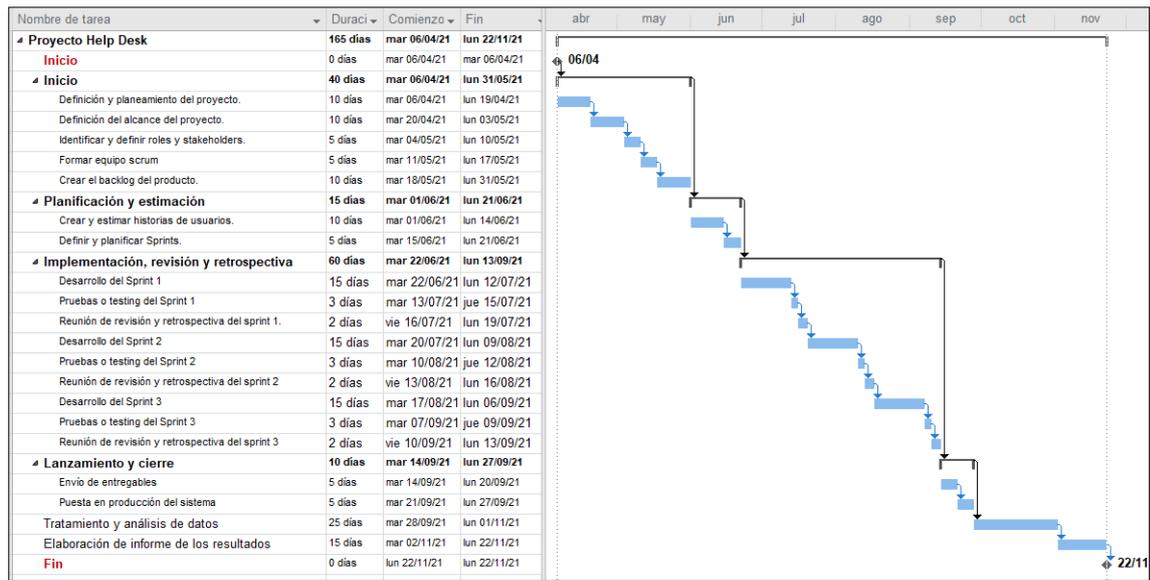


Fig. 4. Cronograma detallado.

3.1.2 SPRINT 01

3.1.2.1 SPRINT BACKLOG

Según el producto backlog o pila del producto, se estableció el primer Sprint (Sprint 1) con las historias de usuario HU01, HU02, HU03 y HU08 (de la tabla XII a la tabla XV), los cuales sirvieron para realizar el backlog del sprint (Tabla XI).

Tabla XI. Backlog del sprint 1.

Historia de Usuario		Tareas	
ID	Deseo...	ID	Descripción
HU01	Gestionar (registrar, listar, editar, eliminar) categorías de incidente, locales municipales y niveles de prioridad de incidentes.	TH1-1	Definir clases categoría, local, prioridad y estado
		TH1-2	Diseñar interfaces y formularios
		TH1-3	Validar interfaces y formularios
		TH1-4	Programar gestión de categorías, locales, prioridad y estado
HU03	Buscar trabajadores y áreas. De acuerdo a su DNI, nombres y apellidos.	TH3-1	Definir tipo de buscador
		TH3-2	Definir condiciones y filtros de búsqueda
		TH3-3	Definir clases a utilizar para el buscador
		TH3-4	Diseñar interfaz o formulario para buscador
		TH3-5	Validar interfaces y formularios de buscador
		TH3-6	Programar buscador
HU02	Crear, editar, eliminar o bloquear un usuario del sistema, además poder asignarle permisos de acceso (rol) dentro del sistema.	TH2-1	Definir roles
		TH2-2	Definir permisos
		TH2-3	Definir clases a utilizar para gestión de usuarios
		TH2-4	Diseñar interfaz y formularios para gestión de usuarios
		TH2-5	Validar interfaces y formularios para gestión de usuarios
		TH2-6	Programar gestión de usuarios
HU08	Iniciar sesión en el sistema	TH8-1	Diseñar interfaz de inicio de sesión
		TH8-2	Validar interfaces y formularios de inicio de sesión
		TH8-3	Definir clases a utilizar para inicio de sesión
		TH8-4	Programar inicio de sesión

3.1.2.2 DESARROLLO DEL SPRINT

Para el desarrollo del Sprint se distribuyeron por historias de usuario, ya que Scrum indica que se debe tratar en lo posible de realizar historias de usuario independientes.

3.1.2.2.1 HISTORIA DE USUARIO HU01

Tabla XII. Historia de usuario "Gestión de categorías, locales y nivel de prioridad".

Historia de usuario	
Código: HU01	Usuario: Supervisor (Supervisor del área de soporte y tecnologías de información)
Nombre de la historia: Gestión de categorías, locales, estados y nivel de prioridad.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Baja
Iteración asignada: Sprint 01	Dependencia para su desarrollo: Ninguna
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Supervisor. Quiero: Gestionar (registrar, listar, editar, eliminar) categorías de incidente, locales municipales, estados y niveles de prioridad de incidentes. Para: Para identificar, categorizar y conocer donde ha ocurrido la incidencia y programar la atención requerida según su nivel de prioridad y categoría.	
Criterios de aceptación: <ul style="list-style-type: none">- El sistema registra el nombre de la categoría, local, estado y nivel de prioridad.- El sistema muestra un apartado independiente para categoría, local, estado y nivel de prioridad.- El sistema actualiza el nombre de la categoría, local, estado y nivel de prioridad.- El sistema elimina el nombre de la categoría, local, estado y nivel de prioridad.- El sistema muestra la lista de categorías, estados, locales y niveles de prioridad.	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El Sistema Help Desk debería de registrar el nombre de la categoría, local, estado y nivel de prioridad.
- El Sistema Help Desk debería mostrar un apartado independiente para categoría, local, estado y nivel de prioridad.
- El Sistema Help Desk debería actualizar el nombre de la categoría, local, estado y nivel de prioridad.
- El Sistema Help Desk debería eliminar el nombre de la categoría, local, estado y nivel de prioridad.
- El Sistema Help Desk debería mostrar la lista de categorías, estados, locales y niveles de prioridad.

Diagrama de clases: en la figura Fig.5 se muestra el diagrama de clases del backlog del producto. Que posteriormente se va desarrollando en cada Sprint.

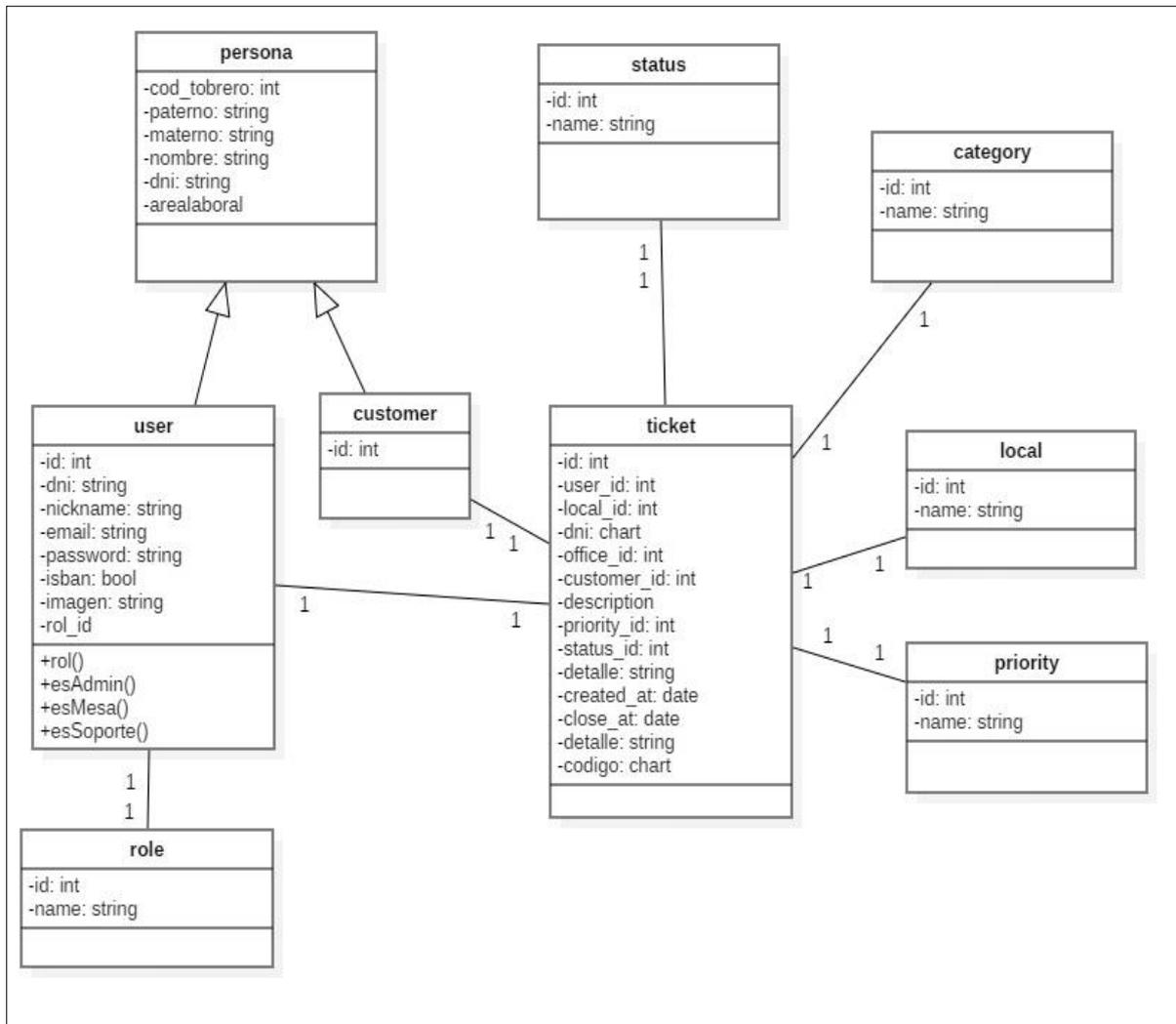


Fig. 5. Diagrama de clases del proyecto.

A continuación, se muestra el diagrama de clases correspondiente a la historia de usuario HU01. Es importante indicar que la clase “model” es propia del framework laravel, la cual es una clase abstracta que contiene una serie de atributos y métodos para las clases hijas. Sin embargo, para fines explicativos se mostraría de la siguiente manera:

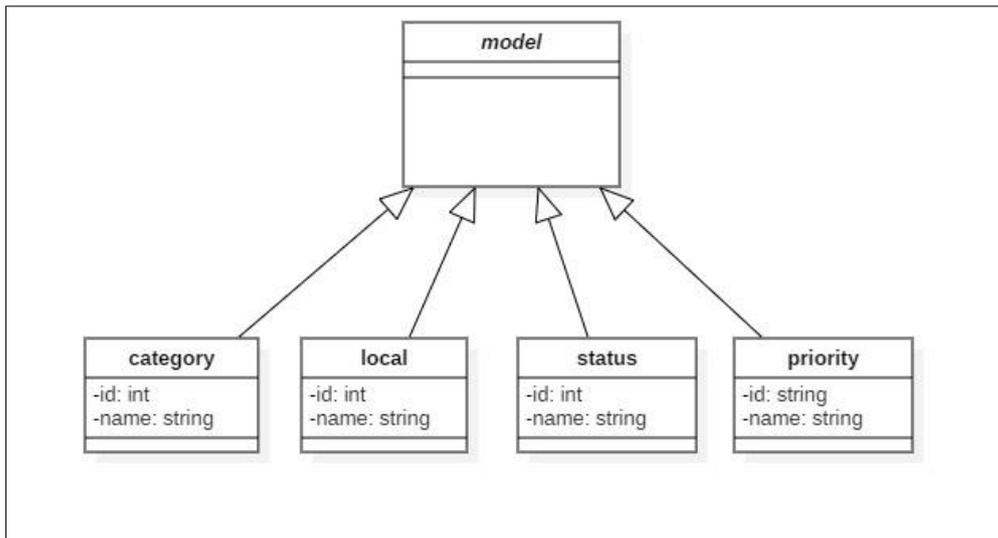


Fig. 6. Diagrama de clases de la HU01

Luego de establecido el diagrama de clases, se procede a crear el proyecto en Laravel. Se crea la clase modelo, controlador y la vista. En este caso, se muestra el código referente a categorías.

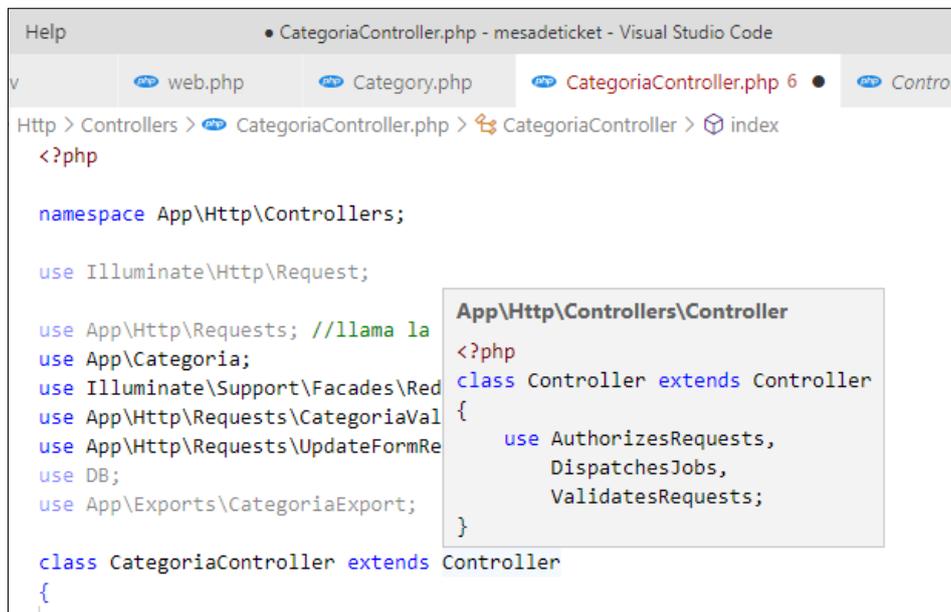
Código fuente: a continuación, se muestra el código referente al desarrollo de la historia de usuario HU01, donde, se trabaja el código del modelo, controlador y vista de categorías:

Código del modelo "Category":

```

<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Category extends Model
{
    protected $table = 'category';
    protected $primaryKey = 'id';
    public $timestamps=false;
    protected $fillable =['name'];
    protected $guarded = [];
}
  
```

Del mismo modo, al crear una clase controlador se extiende o hereda de “Controller”, la cual es propia de Laravel, como se muestra en la siguiente imagen:



```
Help • CategoriaController.php - mesadeticket - Visual Studio Code
v web.php Category.php CategoriaController.php 6 Control
Http > Controllers > CategoriaController.php > CategoriaController > index
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Http\Requests; //llama la
use App\Category;
use Illuminate\Support\Facades\Redirect;
use App\Http\Requests\CategoryValidRequest;
use App\Http\Requests\UpdateFormRequest;
use DB;
use App\Exports\CategoryExport;

class CategoriaController extends Controller
{

```

Fig. 7. Clase "controller".

Por lo que el código del controlador “CategoryController” es el siguiente:

```
<?php
namespace App\Http\Controllers;

class CategoryController extends Controller
{
    public function index()//Request $request
    {
        $categorias = Categoria::all();
        return view('almacen.categoria.index',['categorias'=>$categorias]);
    }
    public function create()
    {
        return view("almacen.categoria.create");
    }
    public function store(CategoryValidRequest $request)
    {
        $categoria = new Categoria;
        $categoria->name=$request->get('name');
        $categoria->save();
        session()->flash('message','Agregado correctamente!');
        return Redirect::to('almacen/categoria');
    }
    public function show($id)
    {

```

```

        return view("almacen.categoria.show",["categoria"=>Categoria::findOrFail($id)];
    }
    public function edit($id)
    {
        return view("almacen.categoria.edit",["categoria"=>Categoria::findOrFail($id)];
    }
    public function update(UpdateFormRequest $request, $id)
    {
        $categoria = Categoria::findOrFail($id);
        $categoria->name=$request->get('name');
        $categoria->update();
        session()->flash('message', 'Actualizado correctamente!');
        return Redirect::to('almacen/categoria');
    }
    public function destroy($id)
    {
        try {
            $categoria = Categoria::findOrFail($id);
            $categoria->delete();

        } catch(\Illuminate\Database\QueryException $e) {
            return Redirect::to('almacen/categoria')->with('error', 'No se puede eliminar el dato!,
Está siendo usado actualmente');
        }
        session()->flash('message', 'Eliminado correctamente!');
        return Redirect::to('almacen/categoria');
    }
}

```

Código de la vista "nueva categoría":

```

@extends ('layouts.admin')
@section('titulo')
<h3>Nueva Categoría</h3>
@endsection
@section ('contenido')
    <div class="row">
        <div class="col-lg-8 col-md-10 col-sm-10 col-xs-12">
            @include('partials.message')
            <!--formulario con etiquetas dinamicas -->
            {!!Form::open(array('url'=>'almacen/categoria','method'=>'POST','autocomplete'=>'off'))!
!}

            {{Form::token()}}
            <div class="form-group">
                <label for="nombre">Nombre o descripción</label>
                <input type="text" name="name" class="form-control" value="{{old('name')}}"
placeholder="Ingrese nombre..." required>
                <br>
            </div>

```

```
<button class="btn btn-primary" type="submit">Guardar</button>
<a href="{route('categoria.index')}" class="btn btn-danger"> Cancelar</a>
{!!Form::close()!!}
</div>
</div>
@endsection
```

Vista del sistema: en las figuras fig.8 y fig.9 se muestra el resultado del desarrollo del código del diagrama de clases de categorías:



Fig. 8. Vista de "nueva categoría".



ORDEN	DESCRIPCION	OPCIONES
1	INTERNET	[Edit] [Delete]
2	HARDWARE	[Edit] [Delete]
3	SOFTWARE	[Edit] [Delete]
4	SIGA	[Edit] [Delete]
5	SIAF	[Edit] [Delete]
6	REDES Y CABLEADO	[Edit] [Delete]

Fig. 9. Vista del sistema opción "listar categorías".

3.1.2.2.2 HISTORIA DE USUARIO HU03

Tabla XIII. Historia de usuario "buscar trabajador".

Historia de usuario	
Código: HU03	Usuario: Encargado de la mesa de ayuda
Nombre de la historia: Buscar trabajador.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Medio
Iteración asignada: Sprint 01	Dependencia para su desarrollo: Ninguna
Programador responsable: David Vásquez Cotrina	
Descripción:	
<p>Como: Encargado de la mesa de ayuda.</p> <p>Quiero: Buscar trabajadores y áreas según DNI, nombres y apellidos.</p> <p>Para: Para identificar al trabajador quien reporta el incidente y generar el ticket de atención.</p>	
Criterios de aceptación:	
<ul style="list-style-type: none"> - El sistema permite busca un trabajador por DNI. - El sistema permite busca un trabajador por nombres y apellidos. 	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería permitir busca un trabajador por DNI.
- El sistema debería permitir buscar un trabajador por nombres y apellidos.

Diagrama de clases: en la fig.10 se muestra el diagrama de clases correspondiente a la historia de usuario HU03, donde se solicita que el sistema permita realizar búsquedas automáticas de los trabajadores quienes van a reportar el incidente:

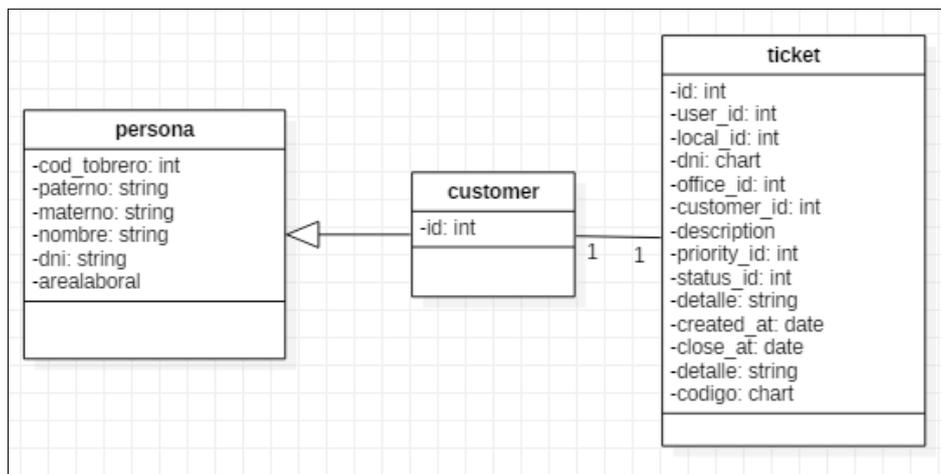


Fig. 10. Diagrama de clases de la HU03.

Prototipo: en la figura fig.11 se muestra el prototipo de la historia de usuario HU03, cabe indicar que, en este punto, sólo se muestra exclusivamente la funcionalidad búsqueda:

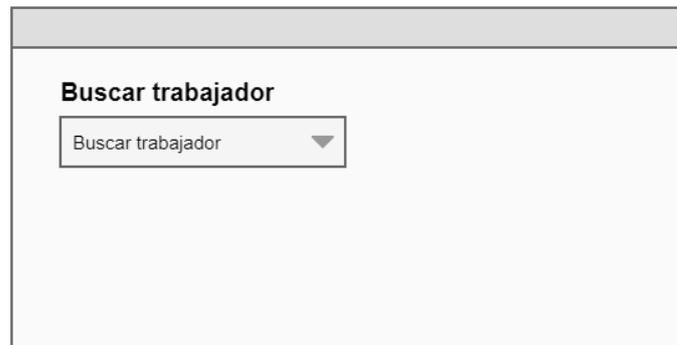


Fig. 11. Prototipo "buscar trabajador".

Código fuente: a continuación, se muestra el código referente a la HU03. En este punto, se agregó un método "consultararea".

Código de la clase modelo:

```
<?php
namespace App;

use Illuminate\Database\Eloquent\Model;

class Customer extends Model
{
    protected $table = 'roles';
    protected $primaryKey = 'id';
    public $timestamps=false;
    protected $fillable =['name'];
    protected $guarded = [];
}
```

Código del controlador:

```
<?php
namespace App\Http\Controllers;
use App\Http\Requests\TicketFormRequest;
use Illuminate\Support\Facades\Redirect;

class CustomerController extends Controller
{
    public function consultararea($id)
    {
        $mytime = Carbon::now('America/Lima'); //zona horaria
        $fecha_cotrato = $mytime->toDateTimeString();
        $area_mysql = DB::connection('mysql2')->table('personal_empresa as pe')
        ->join('tobrero as ob','ob.cod_tobrero','=', 'pe.cod_tobrero')
```

```

->join('x_areas as ar','ar.id','=','pe.areaxx_id')
->select('ar.id','ar.nombre','ob.dni','ob.nombre as obrero','ob.paterno',
        'ob.materno')
->where('pe.cod_tobrero','=',$id) //
->where('pe.fecha_fin','=',null)
->orWhere('pe.cod_tobrero','=',$id) //
->where('pe.fecha_fin','>=',$fecha_cotrato)
->get();
return response()->json(['data' => $area_mysql]);
}

```

Código de la vista para el buscador:

```

{{Form::token()}}
<div class="row">
  <div class="col-lg-6 col-sm-12 col-md-6 col-xs-12">
    <div class="form-group">
      <label>Solicitante:</label>
      <select name="solicitante" class="form-control selectpicker" id="solicitante" data-
live-search="true">
        <option value="">Seleccionar...</option>
        @foreach ($cliente as $cli)
          <option value="{{ $cli->cod_tobrero }}">{{ $cli->dni.' - '.$cli->paterno.' '.$cli-
materno.' '.$cli->nombre}}</option>
        @endforeach
      </select>
    </div>
  </div>
  <div class="col-lg-6 col-sm-6 col-md-6 col-xs-12">
    <div class="form-group">
      <label>Oficina:</label>
      <input type="text" id="areanombre" class="form-control" disabled>
      <input type="text" id="oficina" name="oficina" hidden>
      <input type="text" id="code" name="code" hidden>
    </div>
  </div>
</div>
<hr>
@endsection
@push('scripts')
  <script>
    $('#solicitante').on('change',function() {
      codigo = $(this).val();
      $.ajax({
        url:"{{ url('consultar_area') }}" + codigo,
        success:function(resp){
          $('#areanombre').val(resp.data[0].nombre);
          $('#oficina').val(resp.data[0].id);
          $('#code').val(resp.data[0].dni);
        }
      });
    });
  </script>
@endpush

```

```

    }
  });
});
</script>
@endpush

```

Vista del sistema: en la figura fig.12 se muestra la vista o el resultado de la historia de usuario HU03:



Fig. 12. Vista del sistema "buscador".

3.1.2.2.3 HISTORIA DE USUARIO HU02

Tabla XIV. Historia de usuario "gestión de usuarios".

Historia de usuario	
Código: HU02	Usuario: Supervisor (Supervisor del área de soporte y tecnologías de información)
Nombre de la historia: Gestión de usuarios.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Medio
Iteración asignada: Sprint 01	Dependencia para su desarrollo: Ninguna
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Supervisor. Quiero: Crear, editar, eliminar o bloquear un usuario del sistema, además poder asignarle permisos de acceso (rol) dentro del sistema. Para: Para asociar la atención de un ticket con un responsable para su atención.	
Criterios de aceptación: <ul style="list-style-type: none"> - El sistema registra un usuario. - El sistema muestra un listado de usuarios registrados. - El sistema actualiza un usuario. - El sistema elimina y/o bloquea un usuario. - El sistema permite asignar un rol a un usuario. 	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería registrar un usuario.
- El sistema debería mostrar un listado de usuarios registrados.
- El sistema debería actualizar un usuario.
- El sistema debería eliminar y/o bloquear un usuario.
- El sistema debería asignar un rol a un usuario.

Prototipos: en las figuras fig.13 y fig.14 se muestra los prototipos correspondientes a la historia de usuario HU02.

Prototipo de la interfaz para listar usuarios. El título es "Usuarios". Hay un botón "Agregar Usuario". A continuación hay una tabla con tres columnas: "#", "Usuario" y "Opciones".

#	Usuario	Opciones
1	Usuario 1	Editar Eliminar Bloqu.
2	Usuario 2	Editar Eliminar Bloqu.
3	Usuario 3	Editar Eliminar Bloqu.

Fig. 13. Prototipo "listar usuarios".

Prototipo de la interfaz para agregar un nuevo usuario. El título es "Nuevo Usuario". Hay un menú desplegable "Usuarios disponibles". Hay tres campos de texto: "DNI", "NOMBRES" y "APELLIDOS". Hay un campo de texto "CONTRASEÑA" y un menú desplegable "ROL". Hay dos botones: "GUARDAR" y "CANCELAR".

Fig. 14. Prototipo "agregar nuevo usuario".

Diagrama de clases: en la figura fig.15 se muestra el diagrama de clases correspondiente a la historia de usuario HU02.

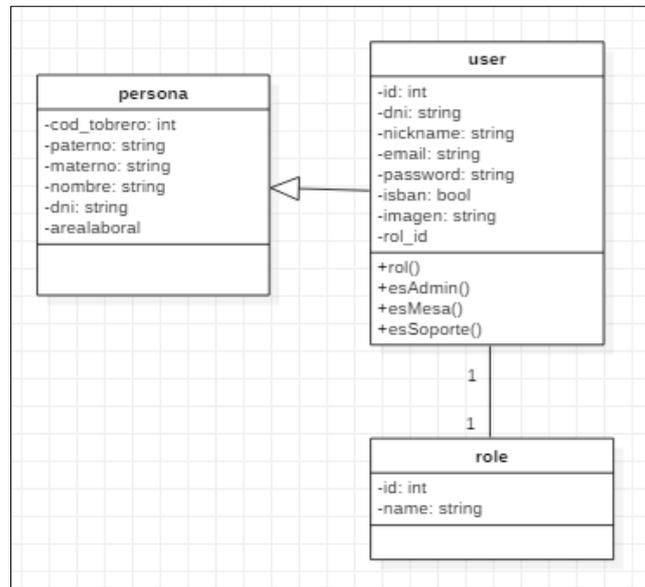


Fig. 15. Diagrama de clases HU02.

Código fuente: a continuación, se muestra partes del código fuente correspondiente a la historia de usuario HU02, donde se realiza un CRUD (crear, listar, actualizar y eliminar) de usuarios del Sistema Help Desk. La clase “Authenticatable” es propio de Laravel, la cual sirve para ayudar a crear el login.

Código del modelo "User":

```

<?php
namespace App;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use Notifiable;
    protected $table = 'users';
    protected $primary_key = 'id';

    protected $fillable = [
        'password', 'dni', 'rol_id'
    ];
    protected $hidden = [
        'password', 'remember_token',
    ];
}

```

```

public function rol(){
    return $this->belongsTo(Role::class,'rol_id');
}

public function esAdmin(){
    if ($this->rol->name == 'ADMINISTRADOR') {
        return true;
    }
    return false;
}

public function esMesa(){
    if ($this->rol->name == 'HELP DESK') {
        return true;
    }
    return false;
}

public function esSoporte(){
    if ($this->rol->name == 'SOPORTE') {
        return true;
    }
    return false;
}
}
}

```

Código del controlador de usuario:

```

<?php
namespace App\Http\Controllers;
use App\User;

class UsuarioController extends Controller
{
    public function index()
    {
        $usuarios_mysql = DB::table('users as u')
        ->join('rrhh.tobrero as ob','u.dni','=','ob.dni')
        ->join('roles as r','r.id','=','u.rol_id')
        ->select('ob.cod_tobrero',
            'ob.dni','ob.nombre','ob.paterno','ob.materno','u.email',
            'u.isban','u.dni','r.name','u.rol_id','u.id','u.created_at'
            )
        ->get();
        return view('admin.usuarios.index',['usuarios'=>$usuarios_mysql]);
    }
    public function create()
    {
        $mytime = Carbon::now('America/Lima');
        $fecha_cotrato = $mytime->toDateTimeString();
        $cliente_mysql = DB::connection('mysql2')->table('personal_empresa as pe')

```

```

->join('tobrero as ob','ob.cod_tobrero','=','pe.cod_tobrero')
->join('x_areas as ar','ar.id','=','pe.areaxx_id')
->select('ob.cod_tobrero', 'ob.dni', 'ob.nombre', 'ob.paterno', 'ob.materno')
->where('ar.id','=','22')
->where('pe.fecha_fin','=','null')
->orWhere('pe.fecha_fin','>=',$fecha_cotrato)
->where('ar.id','=','22')
->get();
$roles = DB::table('roles')->get();
$password = Str::random(6);
return
view("admin.usuarios.create",["pass"=>$password,"cliente"=>$cliente_mysql,"roles"=>$roles]);
}
public function store(UsuarioFormRequest $request)
{
    $usuarios = new User;
    $usuarios->dni = $request->get('dni');
    $usuarios->nickname = $request->get('nombre');
    $usuarios->rol_id = $request->get('rol');
    $usuarios->password = bcrypt($request->get('password'));
    $usuarios->save();
    session()->flash('message', 'Agregado correctamente!');
    return Redirect::to('almacen/usuarios');
}
public function edit($id)
{
    $roles = DB::table('roles')->get();
    $usuarios = DB::table('rrhh.tobrero as ob')
->join('users as u','ob.dni','=','u.dni')
->join('roles as r','r.id','=','u.rol_id')
->select('ob.cod_tobrero',
        'ob.dni', 'ob.nombre', 'ob.paterno', 'ob.materno', 'u.email', 'u.id',
        'r.id as rol')
->where('u.id','=',$id)
->get();
return view("admin.usuarios.edit",["usuarios"=>$usuarios,"roles"=>$roles]);
}
public function update(Request $request, $id)
{
    $email = $request->get('email');
    if ($email) {
        $this->validate($request,array(
            'email'=> "max:255|unique:users,email,$id"
        ));
    }
    $usuarios = User::findOrFail($id);
    $usuarios->rol_id = $request->get('rol_id');
    $usuarios->email = $request->get('email');
    $password = $request->get('password');

```

```

        if ($password) {
            $usuarios->password = bcrypt($request->get('password'));
        }
        $usuarios->save();
        session()->flash('message', 'Actualizado correctamente!');
        return Redirect::to('almacen/usuarios');
    }
    public function destroy($id)
    {
        $us = Auth::user()->id;
        if ($id == $us) {
            return redirect()->back()->with('error', 'Ups! No puedes eliminarte a ti mismo.');
```

```

    }
}

```

Código para la vista "agregar nuevo usuario":

```
@section('titulo')
<div class="col-lg-8 col-md-8 col-dm-8 col-xs-12">
  <h3>Nuevo Usuario</h3>
</div>
@endsection
@section('contenido')
<div class="row">
  <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
    @include('partials.errors')
  </div>
</div>
{!!Form::open(array('url'=>'almacen/usuarios','method'=>'POST','files'=>true,'autocomplete'=>'off'))
!!}
{{Form::token()}}
<div class="row">
  <div class="col-lg-12 col-sm-12 col-md-12 col-xs-12">
    <div class="form-group">
      <label>Usuarios Disponibles:</label>
      <select name="customer_id" class="form-control selectpicker" id="customer_id" data-live-
search="true">
        <option value="">Seleccionar...</option>
        @foreach ($cliente as $cli)
          <option value="{{ $cli->cod_tobrero }}">
            {{ $cli->dni.' - '.$cli->paterno.' '.$cli->materno.' '.$cli->nombre }}</option>
        @endforeach
      </select>
    </div>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-lg-6 col-sm-12 col-md-6 col-xs-12">
    <label for="dni">DNI:</label>
    <input type="text" id="dni" value="{{old('dni')}}" class="form-control" placeholder="DNI"
disabled>
    <input type="text" id="dnii" name="dni" value="{{old('dni')}}" hidden>
    <label for="dni">Nombre:</label>
    <input type="text" id="nombre" name="inten" value="{{old('nombre')}}" class="form-control"
placeholder="NOMBRES" disabled>
    <input type="text" id="nombree" name="nombre" value="{{old('nombre')}}" hidden>
    <label for="dni">Apellidos:</label>
    <input type="text" id="app" name="app" value="{{old('appe')}}" class="form-control"
placeholder="Apellidos" disabled>
    <input type="text" id="appe" name="appe" value="{{old('appe')}}" hidden>
  </div>
</div>
<div class="col-lg-6 col-sm-12 col-md-6 col-xs-12">
```

```

<div class="form-group">
  <label for="dni">Contraseña:</label>
  <input id="password" type="text" class="form-control @error('password') is-invalid
@enderror"
    name="password" value="{{ old('password', '$pass') }}" autocomplete="new-password">
  <label>Rol</label>
  <select name="rol" class="form-control selectpicker" id="rol" data-live-search="true">
    <option value="">Seleccionar...</option>
    @foreach ($roles as $rol)
    <option value="{{ $rol->id }}">{{ $rol->name }}</option>
    @endforeach
  </select>
  <br><br>
  <button class="btn btn-primary" type="submit">Guardar</button>
  <a href="{{ route('usuarios.index') }}" class="btn btn-secondary"> Cancelar</a>
</div>
</div>
{!!Form::close()!!}
@endsection

```

Vistas del sistema: en la figura fig.16 y fig.17 se muestra las vistas del sistema correspondiente a la historia de usuario HU01.

Fig. 16. Vista "nuevo usuario".

#	USUARIO	NOMBRE	CORREO	ROL	ESTADO	OPCIONES
1	43898997	MOISES ISAAC SAUCEDO ROJAS		SOPORTE	Habilitado	  
2	47346268	DAVID VASQUEZ COTRINA		ADMINISTRADOR	Habilitado	  

Fig. 17. Vista de "listar usuarios".

3.1.2.2.4 HISTORIA DE USUARIO HU08

Tabla XV. Historia de usuario "inicio de sesión".

Historia de usuario	
Código: HU08	Usuario: Encargado de la mesa de ayuda
Nombre de la historia: Inicio de sesión.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Medio
Iteración asignada: Sprint 01	Dependencia para su desarrollo: Ninguna
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Supervisor. Quiero: Iniciar sesión el sistema. Para: Para interactuar con las funciones del sistema de acuerdo a mi rol.	
Criterios de aceptación: <ul style="list-style-type: none">- El sistema solicita usuario y contraseña.- El sistema muestra un mensaje de error cuando los datos son incorrectos.- El sistema ingresa y muestra una interfaz de acuerdo al rol del usuario (Administrador, helpdesk y soporte).	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería solicitar usuario y contraseña.
- El sistema debería mostrar un mensaje de error cuando los datos son incorrectos.
- El sistema debería ingresar y mostrar una interfaz de acuerdo al rol del usuario (Administrador, helpdesk y soporte).

En relación a la historia de usuario HU08, se desarrolla el inicio de sesión, para lo cual se utiliza el "login" de Laravel, lo cual viene incluido dentro del framework.

Prototipos: en la figura fig.18 se muestra el prototipo correspondiente a la historia de usuario HU08.

El prototipo muestra una interfaz de usuario con un título "Iniciar Sesión" en un encabezado gris. En el centro hay un cuadro con una 'X' y el texto "100 x 100". Debajo de esto hay un campo de texto con el placeholder "text", un campo de contraseña con "*****" y un botón "Ingresar" en la parte inferior.

Fig. 18. Prototipo de "inicio de sesión".

Diagrama de clases: en la figura fig.19 se muestra el diagrama de clases correspondiente a la historia de usuario HU08.

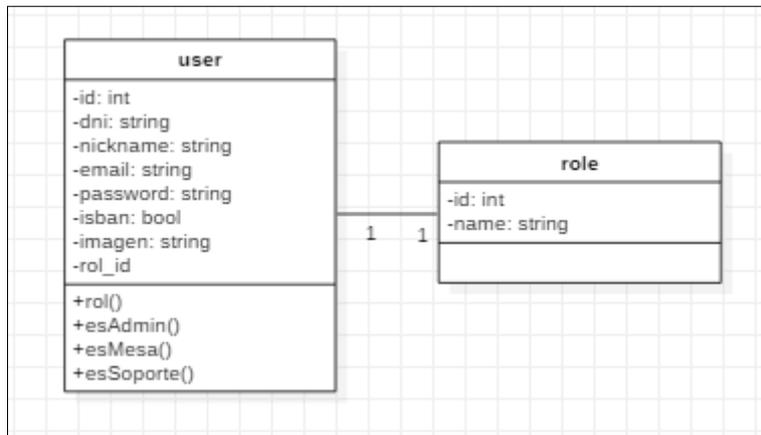


Fig. 19. Diagrama de clases "inicio de sesión".

Código fuente: A continuación, se muestra el parte del código fuente correspondiente a la historia de usuario HU08.

Código del modelo "user":

```
<?php
namespace App;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;

class User extends Authenticatable
{
    use Notifiable;
    protected $table = 'users';
    protected $primary_key = 'id';

    protected $fillable = [
        'nickname', 'email', 'password', 'isban', 'imagen', 'dni', 'rol_id'
    ];
    protected $hidden = [
        'password', 'remember_token',
    ];
    public function rol(){
        return $this->belongsTo(Role::class, 'rol_id');
    }

    public function esAdmin(){
        if ($this->rol->name == 'ADMINISTRADOR') {
            return true;
        }
    }
}
```

```

    }
    return false;
}
public function esMesa(){
    if ($this->rol->name == 'HELP DESK') {
        return true;
    }
    return false;
}
public function esSoporte(){
    if ($this->rol->name == 'SOPORTE') {
        return true;
    }
    return false;
}
}
}

```

Código para el controlador de inicio de sesión:

```

<?php
namespace App\Http\Controllers\Auth;
use Illuminate\Http\Request;
use Illuminate\Foundation\Auth\AuthenticatesUsers;

class LoginController extends Controller
{
    use AuthenticatesUsers;
    protected function redirectTo()
    {
        if (Auth::user()->usertype == 'admin')
        {
            return 'home';
        }
        else
        {
            return 'home';
        }
    }
    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }
    protected function credentials(Request $request)
    {
        $credentials = $request->only($this->username(), 'password');
        return Arr::add($credentials, 'isban', 0);
    }
    public function username()

```

```

{
    return 'dni';
}
}

```

Código para la vista de inicio de sesión:

```

@extends('layouts.app')
@section('content')
<div class="container d-flex justify-content-center">
    <div class="row">
        <div class="card">
            <div class="card-header text-center">
                <strong> {{ __('Sistema Help Desk') }}</strong>
            </div>
            <div class="card-body">
                <form method="POST" action="{{ route('login') }}">
                    @csrf
                    <div class="row text-center">
                        <div class="col">
                            <span class="profile-img">
                                
                            </span>
                        </div>
                    </div>
                    <div class="row">
                        <div class="col">
                            <hr> <!-- other content -->
                        </div>
                    </div>
                    <div class="row">
                        <div class="col">
                            <div class="form-group">
                                <div class="input-group mb-3">
                                    <div class="input-group-prepend">
                                        <span class="input-group-text" id="basic-addon1">
                                            <i class='fas fa-user-shield'></i>
                                        </span>
                                    </div>
                                    <input id="email" type="text"
                                        class="form-control @error('dni') is-invalid @enderror" name="dni"
                                        value="{{ old('dni') }}" required autocomplete="dni" autofocus
                                        placeholder="Usuario">
                                    @error('dni')
                                    <span class="invalid-feedback" role="alert">
                                        </span>
                                    @enderror
                                </div><!-- fin del mb-3 -->
                            </div><!-- form-group -->
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

        <div class="form-group">
          <div class="input-group mb-3">
            <div class="input-group-prepend">
              <span class="input-group-text" id="basic-addon1">
                <i class='fas fa-user-secret'></i>
              </span>
            </div>
            <input id="password" type="password"
              class="form-control @error('password') is-invalid @enderror"
              name="password"
              required autocomplete="current-password" placeholder="Constraseña">
            <span class="invalid-feedback" role="alert">
              @error('password')
            </span>
            @enderror
          </div>
        </div>
        <div class="form-group">
          <button type="submit" class="btn btn-lg btn-info btn-block">
            {{ __('Ingresar') }}
          </button>
        </div>
      </div>
    </div>
  </div>
  </div>
  <hr>
  <link rel='stylesheet' href='https://use.fontawesome.com/releases/v5.5.0/css/all.css'>
  @endsection

```

Vista del sistema: en la figura fig.20 se muestra la vista del sistema correspondiente a la historia de usuario HU08. Donde se aprecia el resultado del desarrollo del inicio de sesión.



Fig. 20. Vista de "inicio de sesión".

3.1.2.3 GRÁFICO BURDOWN

En la siguiente imagen (fig.21) se detalla el proceso de avance del sprint 1, lo cual da como resultado el grafico burdown (fig.22).

Duración del Sprint		15		Esfuerzo Faltante en los días siguientes...																	
Tendencia calculada en los últimos		15	Días	Totales	52	52	46	41	41	39	35	29	29	27	23	19	16	12	5		
ID Tarea	Tarea	ID Historia	Responsable	Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
TH1-1	Definir clases categoría, local, prioridad y esta	HU01	David	Terminado	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH1-2	Diseñar interfaces y formularios	HU01	David	Terminado	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH1-3	Validar interfaces y formularios	HU01	David	Terminado	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH1-4	Programar gestión de categorías, locales, prio	HU01	David	Terminado	8	8	8	3	3	1	0	0	0	0	0	0	0	0	0	0	0
TH3-1	Definir tipo de buscador	HU03	David	Terminado	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0
TH3-2	Definir condiciones y filtros de búsqueda	HU03	David	Terminado	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
TH3-3	Definir clases a utilizar para el buscador	HU03	David	Terminado	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0
TH3-4	Diseñar interfaz o formulario para buscador	HU03	David	Terminado	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0
TH3-5	Validar interfaces y formularios de buscador	HU03	David	Terminado	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
TH3-6	Programar buscador	HU03	David	Terminado	4	4	4	4	4	4	4	4	2	0	0	0	0	0	0	0	0
TH2-1	Definir roles	HU02	David	Terminado	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0
TH2-2	Definir permisos	HU02	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0
TH2-3	Definir clases a utilizar para gestión de usuaric	HU02	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0
TH2-4	Diseñar interfaz y formularios para gestión de i	HU02	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
TH2-5	Validar interfaces y formularios para gestión de	HU02	David	Terminado	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
TH2-6	Programar gestión de usuarios	HU02	David	Terminado	6	6	6	6	6	6	6	6	6	6	6	6	6	2	0	0	0
TH8-1	Diseñar interfaz de inicio de sesión	HU08	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0
TH8-2	Validar interfaces y formularios de inicio de se	HU08	David	Terminado	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
TH8-3	Definir clases a utilizar para inicio de sesión	HU08	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0
TH8-4	Programar inicio de sesión	HU08	David	Terminado	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	0	0

Fig. 21. Backlog detallado del Sprint 1

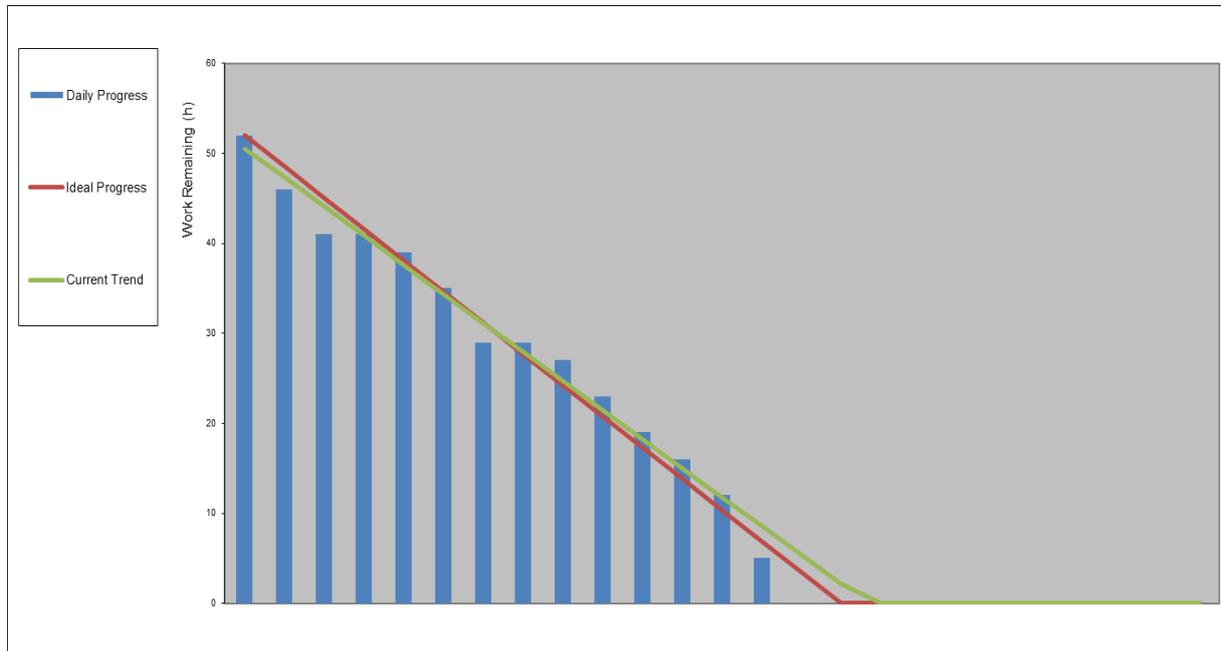


Fig. 22. Gráfico Burdown Sprint 1

3.1.2.4 REVISIÓN, PRUEBAS Y RETROALIMENTACIÓN DEL SPRINT 1

Durante esta ceremonia se revisa el incremento, es decir, lo que se realizó durante el Sprint 1, y se analizan los cambios que tuvo el Backlog del producto. También se mostró al dueño del producto el incremento obtenido después de finalizar el sprint. En la siguiente tabla se detalla los resultados de las pruebas del sprint.

Tabla XVI. Prueba de funcionalidad Sprint 1.

Historia de Usuario	Criterio de aceptación	Resultado esperado	Resultado obtenido
HU01 Gestión de categorías, locales, estados y nivel de prioridad	El sistema registra el nombre de la categoría, local, estado y nivel de prioridad.	Datos almacenados correctamente.	EXITOSO
	El sistema muestra un listado independiente para categoría, local, estado y nivel de prioridad.	<ul style="list-style-type: none"> - El sistema tiene un apartado para categoría, locales, estados y prioridades. - Tiene un menú lateral de navegación. 	EXITOSO
	El sistema actualiza el nombre de la categoría, local, estado y nivel de prioridad.	Datos actualizados correctamente.	EXITOSO
	El sistema elimina el nombre de la categoría, local, estado y nivel de prioridad.	Datos eliminados correctamente.	EXITOSO
	El sistema muestra la lista de categorías, estados, locales y niveles de prioridad.	El sistema muestra la lista de categorías, estados, locales y niveles de prioridad.	EXITOSO
	El sistema debe ser accesible desde un computadora y celular, sin distorsionar su apariencia.	El sistema se ajusta al celular y la computadora de forma dinámica.	EXITOSO
HU03 Buscar trabajador.	El sistema permite buscar un trabajador por DNI.	El sistema muestra una lista desplegable con los resultados	EXITOSO
	El sistema permite buscar un trabajador por nombres y apellidos.	El sistema muestra una lista desplegable con los resultados	EXITOSO
HU02 Gestión de usuarios.	El sistema registra un usuario.	El sistema muestra un mensaje de confirmación.	EXITOSO
	El sistema muestra un listado de usuarios registrados.	Lista de usuarios.	EXITOSO
	El sistema actualiza un usuario.	Datos actualizados correctamente.	EXITOSO
	El sistema elimina y/o bloquea un usuario.	El sistema muestra un mensaje de alerta cuando se intenta eliminar un usuario.	EXITOSO
	El sistema permite asignar un rol a un usuario.	Se muestra una lista desplegable con las opciones disponibles de rol-	EXITOSO

Historia de Usuario	Criterio de aceptación	Resultado esperado	Resultado obtenido
HU08 Inicio de sesión.	El sistema solicita usuario y contraseña.	Se muestra una pantalla para inicio de sesión.	EXITOSO
	El sistema muestra un mensaje de error cuando los datos son incorrectos.	Alerta cuando las credenciales son incorrectas.	EXITOSO
	El sistema ingresar y muestra una interfaz de acuerdo al rol del usuario (Administrador, helpdesk y soporte).	Se muestra la pantalla principal de acuerdo al nivel de usuario logueado.	EXITOSO

3.1.3 SPRINT 02

3.1.3.1 SPRINT BACKLOG

Según el producto backlog o pila del producto, se estableció el segundo sprint (Sprint 2) con las siguientes historias de usuario HU01, HU02, HU03 y HU08 (de la tabla XVIII a la tabla XXI), los cuales sirvieron para realizar el backlog del sprint (Tabla 17).

Tabla XVII. Backlog del sprint 2.

Historia de Usuario		Tareas	
ID	Deseo...	ID	Descripción
HU06	Gestionar (registrar, editar, eliminar, listar) un incidente y asignarle un personal responsable de su atención	TH6-1	Definir clases a utilizar para gestión de incidentes
		TH6-2	Definir campos y tipo de información para registrar incidentes
		TH6-3	Realizar interfaces y formularios
		TH6-4	Validar interfaces y formularios
		TH6-5	Realizar diagrama de clases
		TH6-6	Programar gestión de incidentes
HU07	Gestionar los tickets a mi cargo (listar, cambiar de estado, agregar un comentario).	TH7-1	Definir clases a utilizar
		TH7-2	Definir los accesos cada rol
		TH7-3	Diseñar interfaces y formularios
		TH7-4	Realizar diagrama de clases
		TH7-5	Programar gestión de mis tickets
HU11	Tener un apartado donde poder administrar mi perfil, actualizar mis datos, subir foto y cambiar mi contraseña.	TH11-1	Definir clases a utilizar
		TH11-2	Definir tipo de información a consignar en el perfil
		TH11-3	Diseñar interfaces y formularios
		TH11-4	Realizar diagrama de clases
		TH11-5	Programar administrar perfil
HU04	Tener un dashboard personalizado según el usuario.	TH4-1	Definir clases a utilizar
		TH4-2	Definir tipo de información a consignar en el dashboard
		TH4-3	Diseñar interfaces y formularios
		TH4-4	Realizar diagrama de clases
		TH4-5	Programar dashboard

3.1.3.2 DESARROLLO DEL SPRINT

3.1.3.2.1 Historia de usuario HU06

Tabla XVIII. Historia de usuario "gestión de tickets".

Historia de usuario	
Código: HU06	Usuario: Encargado de la mesa de ayuda
Nombre de la historia: Gestión de tickets.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Media
Iteración asignada: Sprint 02	Dependencia para su desarrollo: HU02
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Encargado de la mesa de ayuda. Quiero: Gestionar (registrar, editar, eliminar, listar) un incidente y asignarle un personal responsable de su atención. Para: Para almacenar toda la información posible del incidente y canalizar su atención.	
Criterios de aceptación: <ul style="list-style-type: none">- El sistema registra el incidente.- El sistema muestra un listado ordenado con los incidentes.- El sistema actualiza o edita el ticket del incidente.- El sistema elimina el ticket de atención.	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería registrar el incidente.
- El sistema debería mostrar un listado ordenado con los incidentes.
- El sistema debería actualizar o editar el ticket del incidente.
- El sistema debería eliminar el ticket de atención.

Prototipos: en la figura fig.23 y fig.24 se muestra los prototipos correspondientes a la historia de usuario HU06.

#	Solicitante	Oficina	Incidente	Hora	Estado	Opciones
1	Trabajador 1	Oficina 1	Internet	15-02-21 09:20		Editar Eliminar
2	Trabajador 2	Oficina 2	Internet	15-02-21 09:20		Editar Eliminar

Fig. 23. Prototipo "listar tickets".

Agregar nuevo ticket

Solicitante:

Oficina:

Descripción del incidente:

Asignado a:

Prioridad:

Estado:

Local:

Categoría:

Fig. 24. Prototipo "agregar nuevo ticket".

Diagrama de clases: en la figura fig.25 se muestra el diagrama de clases correspondiente a la historia de usuario HU06.

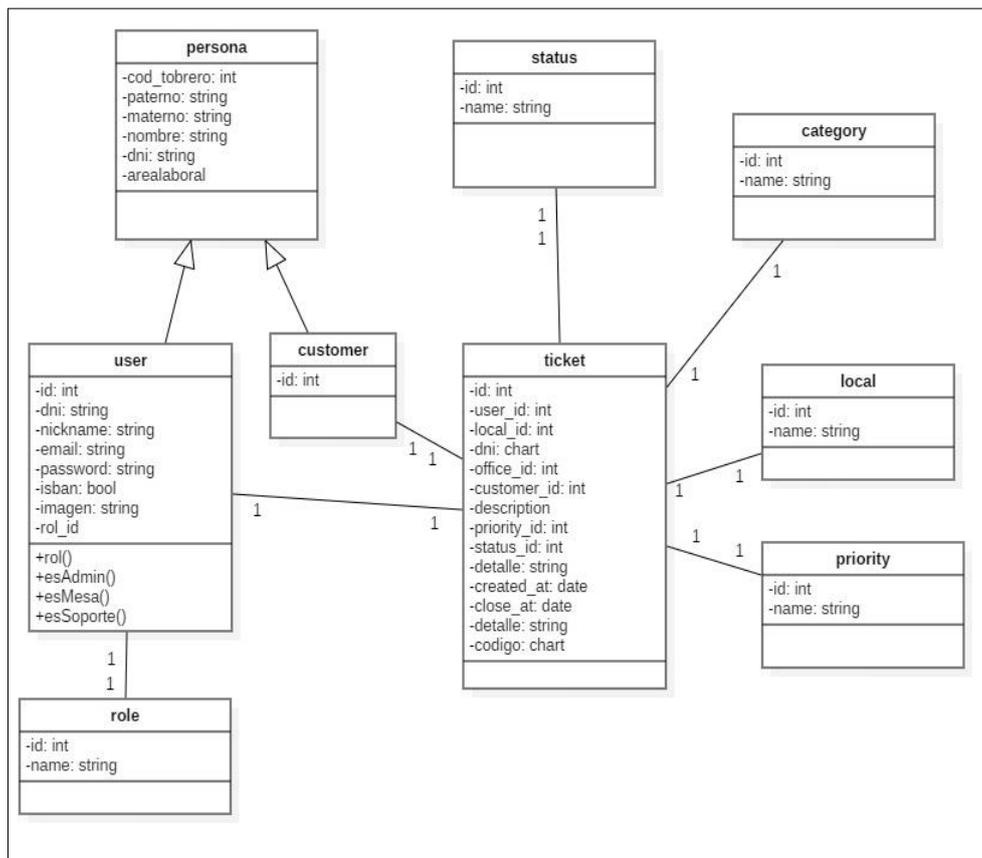


Fig. 25. Diagrama de clases de HU06.

Código fuente: a continuación, se muestra parte del código fuente correspondiente a la historia de usuario HU06, donde se desarrolla, todo lo relacionado al registro de incidentes.

Código del modelo “Ticket”:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

class Ticket extends Model
{
    protected $table = 'ticket';
    protected $primaryKey = 'id';
    public $timestamps=false;
    protected $fillable=[
        'user_id','local_id','office_id','customer_id','description','priority_id',
        'category_id','status_id','detalle','created_at','close_at','codigo',
    ];
    protected $guarded = [];
}
```

Código del controlador de “Ticket”:

```
<?php
namespace App\Http\Controllers;

class TicketController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
    public function index()
    {
        $tickets = DB::table('ticket as t')
            ->join('rrhh.tobrero as ob','ob.cod_tobrero','=','t.customer_id')
            ->join('rrhh.x_areas as ar','ar.id','=','t.office_id')
            ->join('users as u','t.user_id','=','u.id')
            ->join('local as lo','t.local_id','=','lo.id')
            ->join('category as ca','t.category_id','=','ca.id')
            ->join('priority as pr','t.priority_id','=','pr.id')
            ->join('rrhh.tobrero as obr','obr.dni','=','u.dni')
            ->select('t.id' ,
                't.description as title','obr.nombre as user','t.created_at as fecha',
                't.codigo as codigo','pr.name as nivel','pr.id as niv',
                't.customer_id','t.office_id','ob.nombre as obrero','ob.paterno as paterno',
                'ob.materno as materno','ar.nombre as oficina'
            )
    }
}
```

```

        ->where('t.status_id','=','1')//
        ->where('t.local_id','=','1') //
        ->get();
        return view('almacen.ticket.index',['tickets'=>$tickets]);
    }

    public function create()
    {
        $mytime = Carbon::now('America/Lima'); //zona horaria
        $fecha_cotrato = $mytime->toDateTimeString();
        $usuario = DB::table('users as u')
        ->join('rrhh.tobrero as obr','obr.dni','=','u.dni')
        ->select(
            'u.id',
            DB::raw('CONCAT(obr.nombre," ",obr.paterno," ",obr.materno) AS user'))
        ->where('u.isban','=','false') //
        ->get();
        $local = DB::table('local')->get();
        $category = DB::table('category')->get();
        $priority = DB::table('priority')->get();
        $status = DB::table('status')->get();
        $oficina_mysql = DB::connection('mysql2')->table('x_areas')->get();
        $cliente_mysql = DB::connection('mysql2')->table('personal_empresa as pe')
        ->join('tobrero as ob','ob.cod_tobrero','=','pe.cod_tobrero')
        ->select('ob.cod_tobrero',
            'ob.dni','ob.nombre','ob.paterno','ob.materno')
        ->where('pe.fecha_fin','=','null') //
        ->orWhere('pe.fecha_fin','>=', $fecha_cotrato) //
        ->get();
        return
        view("almacen.ticket.create",["usuario"=>$usuario,"cliente"=>$cliente_mysql,"local"=>$local,"category"=>$category,"oficina"=>$oficina_mysql,"priority"=>$priority,"status"=>$status]);
    }
    public function store(TicketFormRequest $request)
    {
        $ticket = new Ticket();
        $ticket ->dni =$request->get('code');
        $ticket ->description =$request->get('descripcion');
        $ticket ->local_id =$request->get('local');
        $ticket ->user_id =$request->get('asignar'); // ?: null
        $ticket ->office_id =$request->get('oficina');
        $ticket ->category_id =$request->get('category_id');
        $ticket ->customer_id =$request->get('solicitante');
        $ticket ->priority_id =$request->get('prioridad');
        $ticket ->status_id =$request->get('estado');
        $ticket ->detalle =$request->get('detalle');

        $mytime = Carbon::now('America/Lima'); //zona horaria
        $ticket ->created_at = $mytime->toDateTimeString();
    }

```

```

        $codigo = Str::random(4);
        $ticket->codigo = Str::upper($codigo);
        $ticket->save();
        session()->flash('message', 'Agregado correctamente!');
        return Redirect::to('almacen/ticket');
    }
    public function show($id) //es para mostrar
    {
        $ticket = DB::table('ticket as t')
        ->join('rrhh.x_areas as ar','t.office_id','=','ar.id')
        ->join('rrhh.tobrero as ob','t.customer_id','=','ob.cod_tobrero')
        ->join('users as u','t.user_id','=','u.id')
        ->join('local as lo','t.local_id','=','lo.id')
        ->join('status as s','t.status_id','=','s.id')
        ->join('priority as p','t.priority_id','=','p.id')
        ->join('category as ca','t.category_id','=','ca.id')
        ->join('rrhh.tobrero as obr','obr.dni','=','u.dni')
        ->select('t.id',
            DB::raw('CONCAT(obr.paterno," ",obr.materno," ",obr.nombre) AS user'),
            DB::raw('CONCAT(ob.paterno," ",ob.materno," ",ob.nombre) AS nombre'),
            'ob.dni as code','ar.nombre as oficina','t.created_at as fecha','t.description',
            'p.name as priority','s.name as status','lo.name as local','ca.name as
categoryy',
            't.detalle as detallee','t.created_at as open','t.close_at as close','t.codigo as
codd'
        )
        ->where('t.id','=',$id)
        ->first();
        return view('almacen.ticket.show',['ticket'=>$ticket]);
    }
    public function edit($id)
    {
        $mytime = Carbon::now('America/Lima'); //zona horaria
        $fecha_cotrato = $mytime->toDateTimeString();
        $cliente = DB::table('rrhh.personal_empresa as pe')
        ->join('rrhh.tobrero as ob','ob.cod_tobrero','=','pe.cod_tobrero')
        ->select('ob.cod_tobrero',
            'ob.dni','ob.nombre','ob.paterno','ob.materno')
        ->where('pe.fecha_fin','=','null') //contratos inderterminados
        ->orWhere('pe.fecha_fin','>=',$fecha_cotrato) //contratosvigentes
        ->get();

        $oficina = DB::table('rrhh.x_areas as ar')
        ->join('ticket as tik','ar.id','=','tik.office_id')
        ->select('ar.id','ar.nombre as off')
        ->where('tik.id','=',$id)
        ->get();

        $ticket = Ticket::findOrFail($id);
        $usuario = DB::table('users as u')
        ->join('rrhh.tobrero as obr','obr.dni','=','u.dni')

```

```

->select('u.id',DB::raw('CONCAT(obr.nombre," ",obr.paterno," ",obr.materno) AS nickname'))
->where('u.isban','=',false) //solo usuarios no bloqueados
->get();
$local = DB::table('local')->get();
$category = DB::table('category')->get();
$priority = DB::table('priority')->get();
$status = DB::table('status')->get();
return
view("almacen.ticket.edit",["ticket"=>$ticket,"usuario"=>$usuario,"cliente"=>$cliente,"local"=>$local,
"category"=>$category,"oficina"=>$oficina,"priority"=>$priority,"status"=>$status]);
}
public function update(TicketFormRequest $request, $id)
{
    $ticket = Ticket::findOrFail($id);
    $ticket->office_id = $request->get('oficina');
    $ticket->customer_id = $request->get('solicitante');
    $ticket->description = $request->get('descripcion');
    $ticket->user_id = $request->get('asignar');
    $ticket->priority_id = $request->get('prioridad');
    $ticket->status_id = $request->get('estado');
    $ticket->local_id = $request->get('local');
    $ticket->category_id = $request->get('category_id');
    $ticket->detalle = $request->get('detalle');
    $mytime1 = Carbon::now('America/Lima'); //zona horaria
    $ticket ->close_at = $mytime1->toDateTimeString();
    $ticket->update();
    return Redirect::to('almacen/ticket');
}

public function destroy($id)
{
    $ticket = Ticket::findOrFail($id);
    $ticket->delete();
    session()->flash('message','Eliminado correctamente!');
    return redirect()->back();
}

public function externos()
{
    $tickets = DB::table('ticket as t')
    ->join('rrhh.x_areas as ar','t.office_id','=','ar.id')
    ->join('rrhh.tobrero as ob','t.customer_id','=','ob.cod_tobrero')
    ->join('users as u','t.user_id','=','u.id')
    ->join('local as lo','t.local_id','=','lo.id')
    ->join('category as ca','t.category_id','=','ca.id')
    ->join('priority as pr','t.priority_id','=','pr.id')
    ->join('status as s','t.status_id','=','s.id')
    ->join('rrhh.tobrero as obr','obr.dni','=','u.dni')
    ->select('t.id' ,
            't.description as title','obr.nombre as user','ob.nombre as cuname',

```

```

        'ob.paterno as paterno', 'ob.materno as materno', 'ar.nombre as oficina',
        'lo.name as local', 't.created_at as fecha', 't.codigo as codigo',
        'pr.name as nivel', 'pr.id as niv', 's.name as status', 's.id as stat'
    )
    ->where('t.local_id', '!=', '1') //
    ->whereIn('t.status_id', ['1', '2'])//
    ->get();
    return view('almacen.ticket.externos', ["tickets"=>$tickets]);
}
}

```

Código para la vista crear ticket o registrar incidente:

```

@section('titulo')
<div class="col-lg-8 col-md-8 col-sm-12 col-xs-12">
<h3>Agregar Nuevo Ticket</h3>
</div>
@endsection
@section ('contenido')
<div class="row">
    <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
        @if(count($errors)>0)
        <div class="alert alert-danger">
            <ul>
                @foreach ($errors->all() as $error)
                <li>{{$error}}</li>
                @endforeach
            </ul>
        </div>
        @endif
    </div>
    {!!Form::open(array('url'=>'almacen/ticket', 'method'=>'POST', 'autocomplete'=>'off'))!!}
    {{Form::token()}}
    <div class="row">
        <div class="col-lg-6 col-sm-12 col-md-6 col-xs-12">
            <div class="form-group">
                <label>Solicitante:</label>
                <select name="solicitante" class="form-control selectpicker" id="solicitante" data-
live-search="true">
                    <option value="">Seleccionar...</option>
                    @foreach ($cliente as $cli)
                    <option value="{{ $cli->cod_tobrero }}">{{ $cli->dni.' - '.$cli->paterno.' '.$cli-
>materno.' '.$cli->nombre}}</option>
                    @endforeach
                </select>
            </div>
        </div>
    </div>
    <div class="col-lg-6 col-sm-6 col-md-6 col-xs-12">

```

```

    <div class="form-group">
      <label>Oficina:</label>
      <input type="text" id="areanombre" class="form-control" disabled>
      <input type="text" id="oficina" name="oficina" hidden>
      <input type="text" id="code" name="code" hidden>
    </div>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-lg-7 col-sm-8 col-md-7 col-xs-12">
    <div class="form-group">
      <label for="descripcion">Descripción del incidente <em>(400 caracteres max.)</em>:
      </label><!-- old regresa valida request-->
      <textarea type="text" name="descripcion" required class="form-control" id="descripcion"
        rows="3" placeholder="Cuál es el inconveniente?"
        maxlength="400">{{old('descripcion')}}
      </textarea>
      <em><p id="contador"></p></em>
    </div>
  </div>
  <div class="col-lg-5 col-sm-4 col-md-5 col-xs-12">
    <div class="form-group">
      <label>Asignado a:</label>
      <select name="asignar" class="form-control selectpicker" id="asignar" data-live-
        search="true">
        <option value="">Sin asignar...</option>
        @foreach ($usuario as $usu)
        <option value="{{ $usu->id }}">{{ $usu->user }}</option>
        @endforeach
      </select>
    </div>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-lg-2 col-sm-6 col-md-4 col-xs-12">
    <div class="form-group">
      <label class="text-success">Prioridad:</label>
      <select name="prioridad" class="form-control selectpicker" id="prioridad" data-live-
        search="true">
        @foreach ($priority as $pry)
        <option value="{{ $pry->id }}">{{ $pry->name }}</option>
        @endforeach
      </select>
    </div>
  </div>
  <div class="col-lg-2 col-sm-4 col-md-4 col-xs-12">

```

```

    <div class="form-group">
      <label class="text-danger">Estado:</label>
      <select name="estado" class="form-control selectpicker" id="estado" data-live-
search="true">
        @foreach ($status as $stu)
          <option value="{{ $stu->id }}">{{ $stu->name }}</option>
        @endforeach
      </select>
    </div>
  </div>
  <div class="col-lg-4 col-sm-4 col-md-4 col-xs-12">
    <div class="form-group">
      <label>Local:</label>
      <select name="local" class="form-control selectpicker" id="local" data-live-
search="true">
        @foreach ($local as $loc)
          <option value="{{ $loc->id }}">{{ $loc->name }}</option>
        @endforeach
      </select>
    </div>
  </div>
  <div class="col-lg-4 col-sm-4 col-md-3 col-xs-12">
    <div class="form-group">
      <label>Categoría:</label>
      <select name="category_id" class="form-control selectpicker" id="id_categ" data-live-
search="true">
        @foreach ($category as $ctg)
          <option value="{{ $ctg->id }}">{{ $ctg->name }}</option>
        @endforeach
      </select>
    </div>
  </div>
</div>
<hr>
<div class="row">
  <div class="col-lg-12 col-sm-12 col-md-12 col-xs-12">
    <div class="form-group text-center">
      <button class="btn btn-primary" type="submit">Guardar</button>
      <a href="{{ route('ticket.index') }}" class="btn btn-secondary">Cancelar</a>
    </div>
  </div>
</div>
{!! Form::close() !!}
@endsection

```

Vistas del sistema: en la figura fig.26 y fig.27 se muestra vistas del sistema correspondiente a la historia de usuario HU06.

Agregar Nuevo Ticket

Solicitante: Oficina:

Descripción del incidente (400 caracteres max.): Asignado a:

Prioridad: Estado: Local: Categoría:

Fig. 26. Vista de "agregar nuevo ticket".

TICKETS PENDIENTES

Mostrar registros Buscar:

CÓDIGO ↕	FECHA ↕	ASUNTO ↕	SOLICITANTE ↕	OFICINA ↕	NIVEL ↕	OPCIONES ↕
5390	2021-11-25 08:25:04	patrimonio: usuario solicita configuración de impresora	ELADIO SANCHEZ VARGAS	UNIDAD DE LOGISTICA Y SERVICIOS GENERALES SERVICIOS GENERALES	NORMAL	<input type="button" value="✎"/> <input type="button" value="✖"/>

Mostrando del 1 al 1 de un total de 1 registros Anterior Siguiente

Fig. 27. Vista de "listar tickets".

3.1.3.2.2 Historia de usuario HU07

Tabla XIX. Historia de usuario "gestión de mis tickets".

Historia de usuario	
Código: HU07	Usuario: Equipo de soporte
Nombre de la historia: Gestión de mis tickets.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Media
Iteración asignada: Sprint 02	Dependencia para su desarrollo: HU08
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Equipo de soporte. Quiero: Gestionar los tickets a mi cargo (listar, cambiar de estado, agregar un comentario). Para: Para guiarme y atender mis tickets.	
Criterios de aceptación: <ul style="list-style-type: none">- El sistema muestra una lista con mis tickets.- El sistema permite cambiar de estado un ticket seleccionado.- El sistema permite agregar un comentario o descripción de la atención realizada.	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería mostrar una lista de los tickets a cargo del usuario logueado.
- El sistema debería permitir cambiar de estado un ticket seleccionado, de pendiente a atendido o en desarrollo.
- El sistema debería permitir agregar un comentario o descripción de la atención realizada.

Prototipos: en la figura fig.28 y fig.29 se muestran los prototipos correspondientes a la historia de usuario HU07.

#	Solicitante	Oficina	Incidente	Hora	Estado	Opciones
1	Trabajador 1	Oficina 1	Internet	Pendiente	15-02-21 09:20	<input type="button" value="Ver"/>
2	Trabajador 2	Oficina 2	Internet	Pendiente	15-02-21 09:20	<input type="button" value="Ver"/>
3	Trabajador 3	Oficina 3	Internet	Pendiente	15-02-21 09:20	<input type="button" value="Ver"/>

Fig. 28. Prototipo de "mis tickets".

ticket

Condición:

Hora de registro:

Solicitante:

Oficina:

Local:

Detalle del incidente:

Detalle de atención:

Fig. 29. Prototipo de "detalle de ticket"

Diagrama de clases: en la figura fig.29 se muestra el diagrama de clases correspondiente a la historia de usuario HU07.

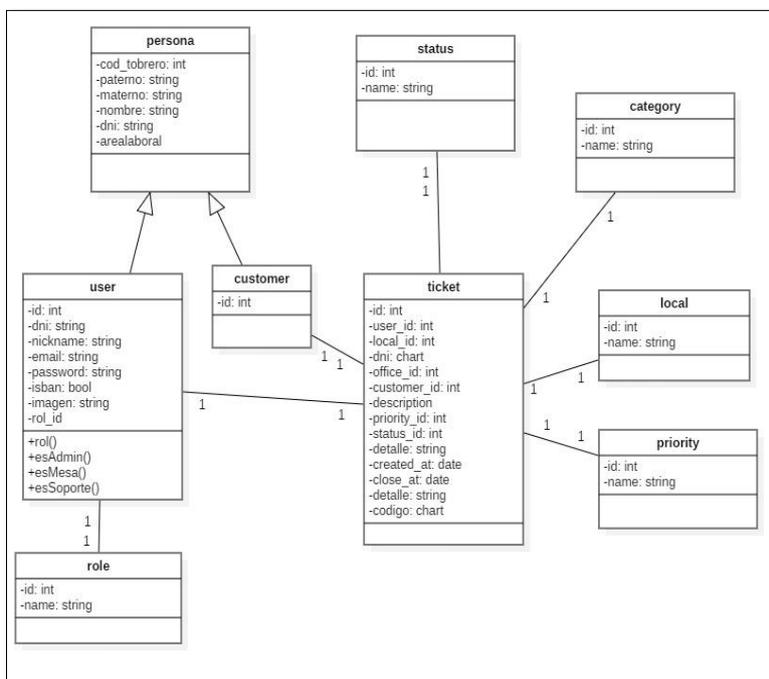


Fig. 30. Diagrama de clases de HU07

Código fuente: en la figura fig.43 se muestra parte del código fuente correspondiente a la historia de usuario HU07. Donde se agregan los métodos: mistickets(), misdesarrollo(), para listar los incidentes a cargo del usuario logueado.

Código de los métodos agregados en el controlador ticket:

```
public function mistickets()
{
    $tickets = DB::table('ticket as t')
        ->join('rrhh.x_areas as ar','t.office_id','=', 'ar.id')
        ->join('rrhh.tobrero as ob','t.customer_id','=', 'ob.cod_tobrero')
        ->join('users as u','t.user_id','=', 'u.id')
        ->join('local as lo','t.local_id','=', 'lo.id')
        ->join('category as ca','t.category_id','=', 'ca.id')
        ->join('status as st','t.status_id','=', 'st.id')
        ->join('priority as p','t.priority_id','=', 'p.id')
        ->select('t.id',
            't.description as title','t.codigo as cod','lo.name as local',
            'ob.nombre as cuname','ob.paterno as paterno','ob.materno as materno','ar.nombre as
            oficina','t.created_at as fecha','p.id as prio','p.name as prioridad'
        )
        ->where('t.status_id','=', '1')
        ->where('t.user_id','=', $us)
        ->get();
    return view('admin.soporte.mistickets',['tickets'=>$tickets]);
}

public function misdesarrollo()
{
    $us = Auth::user()->id;
    $tickets = DB::table('ticket as t')
        ->join('rrhh.x_areas as ar','t.office_id','=', 'ar.id')
        ->join('rrhh.tobrero as ob','t.customer_id','=', 'ob.cod_tobrero')
        ->join('users as u','t.user_id','=', 'u.id')
        ->join('local as lo','t.local_id','=', 'lo.id')
        ->join('category as ca','t.category_id','=', 'ca.id')
        ->join('status as st','t.status_id','=', 'st.id')
        ->select('t.id' ,
            't.codigo as cod','t.description as title','st.name as estado',
            'ob.nombre as cuname','ob.paterno as paterno','ob.materno as materno',
            'ar.nombre as oficina','t.created_at as fecha'
        )
        ->where('t.status_id','=', '2')
        ->where('t.user_id','=', $us)
        ->orderBy('t.id','desc')
        ->get();
    return view('admin.soporte.desarrollo',['tickets'=>$tickets]);
}

public function miticketedit($id)
{
    $ticket = DB::table('ticket as t')
        ->join('rrhh.x_areas as ar','t.office_id','=', 'ar.id')
        ->join('rrhh.tobrero as ob','t.customer_id','=', 'ob.cod_tobrero')
```

```

->join('users as u','t.user_id','=', 'u.id')
->join('local as lo','t.local_id','=', 'lo.id')
->join('status as s','t.status_id','=', 's.id')
->join('priority as p','t.priority_id','=', 'p.id')
->join('category as ca','t.category_id','=', 'ca.id')
->select('t.id',
        DB::raw('CONCAT(ob.nombre," ",ob.paterno," ",ob.materno) AS nombre'),
        'ar.nombre as oficina','t.created_at as fecha','t.description',
        'p.name as priority','s.id as status','lo.name as locall',
        'ca.name as categoryy','t.detalle as detallee','t.created_at as open',
        't.codigo as codd'
        )
->where('t.id','=', $id)
->first();

$status = DB::table('status')->get();

return view("admin.sopORTE.edit",["ticket"=>$ticket,"status"=>$status]);
}
}

```

Código de la vista “mis tickets”:

```

@section('titulo')
<div class="col-lg-12 col-sm-12 col-dm-12 col-xs-12">
    <h3>Mis Tickets Pendientes</h3>
</div>
@endsection
@section('contenido')
<hr>
@include('partials.message')
<div class="row">
    <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <div class="table-responsive">
            <table class="table table-striped table-bordered table-condesed table-hover"
            id="tblticket">
                <thead class="table-warning">
                    <th>COD</th>
                    <th>FECHA</th>
                    <th>ASUNTO</th>
                    <th>SOLICITANTE</th>
                    <th>OFICINA</th>
                    <th>NIVEL</th>
                    <th>LOCAL</th>
                    <th>OPCIONES</th>
                </thead>
                <tbody>
                    @foreach($tickets as $tic)

```

```

        <tr>
            <td>{{ $tic->id.'-'. $tic->cod}}</td>
            <td>{{ $tic->fecha}}</td>
            <td>{{ $tic->title}}</td>
            <td>{{ $tic->paterno.' ' . $tic->materno.' ' . $tic->cuname}}</td>
            <td>{{ $tic->oficina}}</td>
            @if ( $tic->prio=='2')
            <td>
                <span class="btn btn-danger">{{ $tic->prioridad}}</span>
            </td>
            @else
            <td>{{ $tic->prioridad}}</td>
            @endif
            <td>{{ $tic->local}}</td>
            <td>
                <a href="{{URL::action('AjaxController@edit', $tic->id)}}"><button
                    class="btn btn-info fa fa-edit"></button></a>
            </td>
        </tr>
    @endforeach
</tbody>
</table>
</div>
</div>
</div>
@endsection

```

Vistas del sistema: en la figura fig.31 y fig.32 se muestran las vistas del sistema correspondiente a la historia de usuario HU07.

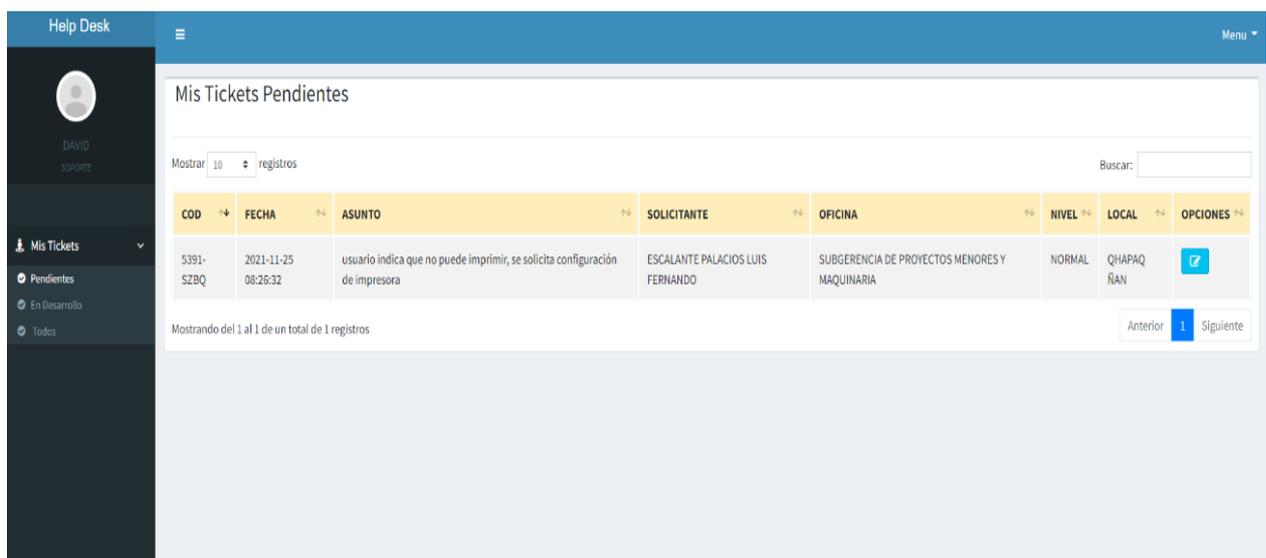


Fig. 31. Vista de "ver mis tickets".

Help Desk Menu

MOISES ISAAC
SOPORTE

Dashboard
Mis Tickets
Calendario

Editar Ticket No: 9-ITJP

CONDICIÓN	PENDIENTE
HORA DE RESISTRO	2022-08-04 21:41:15
SOLICITANTE	JAIME ALBERTO RODRIGO SILVA SANTISTEBAN
OFICINA	GERENCIA DE TURISMO, CULTURA Y CENTRO HISTORICO
CATEGORÍA	HARDWARE
DETALLE DEL INCIDENTE	Se muestra pantalla azul cada 20 min, y se reinicia la pc.
DETALLE DE ATENCIÓN	Detalle de Atención (400 caracteres max.): Cómo lo solucionaste?

Guardar Cancelar

Fig. 32. Vista "detalle de ticket".

3.1.3.2.3 Historia de usuario HU11

Tabla XX. Historia de usuario "perfil de usuario".

Historia de usuario	
Código: HU11	Usuario: Equipo de soporte
Nombre de la historia: Perfil de usuario.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Media
Iteración asignada: Sprint 02	Dependencia para su desarrollo: Ninguno
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Equipo de soporte. Quiero: Tener un apartado donde poder administrar mi perfil, actualizar mis datos, subir foto y cambiar mi contraseña. Para: Para personalizar mi perfil.	
Criterios de aceptación: <ul style="list-style-type: none"> - El sistema muestra los datos del usuario logueado. - El sistema permite cambiar contraseña. - El sistema permite cargar una foto de perfil. 	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería mostrar los datos del usuario logueado.
- El sistema debería permitir cambiar contraseña.
- El sistema debería permitir cargar una foto de perfil.

Prototipos: en la figura fig.33 se muestra el prototipo correspondiente a la historia de usuario HU11.



Fig. 33. Prototipo de "mi perfil".

Diagrama de clases: en la figura fig.46 se muestra el diagrama de clases correspondiente a la historia de usuario HU11.

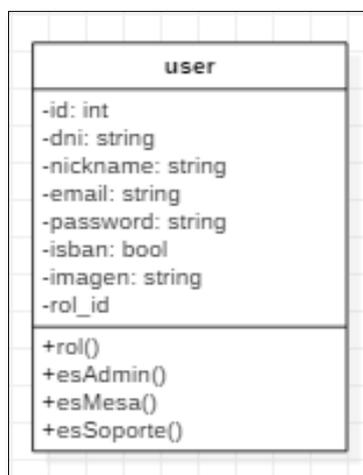


Fig. 34. Diagrama de clases de HU11.

Código fuente: a continuación, se muestra parte del código fuente correspondiente a la historia de usuario HU11. Donde se agrega los métodos miperfil() y updatemiperfil().

Código de los métodos agregados en el controlador usuario:

```
public function miperfil($id)
{
    $usuario = User::findOrFail($id);
    $usuariostotal = DB::table('users as u')
->join('rrhh.tobrero as ob', 'ob.dni', '=', 'u.dni')
->select('u.id as id', 'ob.nombre', 'ob.paterno', 'ob.materno', 'ob.dni')
->where('u.id', '=', $id)
->get();
    return view("admin.sopORTE.miperfil", ["usuarios"=>$usuario, "usuar"=>$usuariostotal]);
}
public function updatemiperfil(Request $request, $id)
{
    $email = $request->get('email');
    if ($email) {
        $this->validate($request, array(
            'email'=> "max:255|unique:users,email,$id",
        ));
    }
    $usuarios = User::findOrFail($id);
    $password = $request->get('password');
    if ($password) {
        $this->validate($request, array('password'=>'min:6'));
        $usuarios->password = bcrypt($request->get('password'));
    }
    if ($request->hasFile('imagen')) {
        $email = Auth::user()->dni;
        $image_path = public_path(). '/imagenes/usuarios/'. $usuarios->email. '/' . $usuarios-
>imagen;
        if (@getimagesize($image_path)) {
            unlink($image_path);
        }
        $file = $request->file('imagen');
        $file->move(public_path(). '/imagenes/usuarios/'. $email, $file->getClientOriginalName());
        $usuarios->imagen = $file->getClientOriginalName();
    }
    $usuarios->save();
    session()->flash('message', 'Actualizado correctamente!');
    return redirect()->back();
}
```

Código de la vista para “mi perfil”:

```
@section ('contenido')
<hr>
<div class="container bootstrap snippet">
  <div class="row">
    <div class="col-sm-10">
      <h1>Mi perfil</h1>
    </div>
  </div>
  {!!Form::model($usuarios,['method'=>'POST','route'=>['miperfil.update'],$usuarios-
>id], 'files'=>true)}!!}
  {{Form::token()}}
  <div class="row">
    <div class="col-lg-6 col-sm-12 col-dm-12 col-xs-12">
      <!--left col-->
      <div class="text-center">
        @if ($usuarios->imagen!="")
          
          <br><br>
        @else
          
          <h6>Agrega una foto...</h6>
        @endif
        <input type="file" id="imagen" name="imagen" class="text-center center-block file-
upload">
      </div>
      <hr><br>
      <div class="panel panel-default text-center">
        <div class="panel-heading">Social Media</div>
        <div class="panel-body">
          <i class="fa fa-facebook fa-2x"></i> <i class="fa fa-github fa-2x"></i> <i
class="fa fa-twitter fa-2x"></i> <i class="fa fa-pinterest fa-2x"></i> <i
class="fa fa-google-plus fa-2x"></i>
        </div>
      </div>
    </div>
    <div class="col-lg-6 col-sm-12 col-dm-12 col-xs-12">
      <div class="tab-content">
        <div class="tab-pane active" id="home">
          <form class="form" action="###" method="post" id="registrationForm">
            <div class="form-group">
              <div class="col-xs-6">
                <label for="first_name">
                  <h4>Usuario</h4>
                </label>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
```

```

        </label>
        <input type="text" class="form-control @error('name') is-invalid
@enderror" name="name" id="first_name" type="text" value="{{ $usuarios->dni }}" disabled>
        @error('name')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>
</div>
<div class="form-group">
    <div class="col-xs-6">
        <label for="lastname">
            <h4>Nombres y Apellidos</h4>
        </label>
        <input type="text" class="form-control @error('lastname') is-invalid
@enderror"
            name="lastname" id="lastname" type="text" value="{{ $usuar[0]-
>nombre.' '. $usuar[0]->paterno.' '. $usuar[0]->materno }}" disabled>
        @error('lastname')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>
</div>
<div class="form-group">
    <div class="col-xs-6">
        <label for="email">
            <h4>Correo electrónico</h4>
        </label>
        <input type="email" class="form-control @error('email') is-invalid
@enderror"
            name="email" id="email" value="{{ $usuarios->email }}"
            autocomplete="email"
            placeholder="you@email.com" title="enter your email.">
        @error('email')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>
</div>
<div class="form-group">
    <div class="col-xs-6">
        <label for="password">
            <h4>Contraseña</h4>
        </label>

```

```

        <input type="password" class="form-control @error('password') is-
invalid @enderror"

        name="password" id="password" value="{{ old('password') }}"
        autocomplete="new-password" placeholder="Mínimo 6 caracteres"
        title="enter your password.">
        @error('password')
        <span class="invalid-feedback" role="alert">
            <strong>{{ $message }}</strong>
        </span>
        @enderror
    </div>
</div>
<div class="form-group">
    <div class="col-xs-12">
        <br>
        <button class="btn btn-xs btn-primary" type="submit"><i
            class="glyphicon glyphicon-ok-sign"></i> Guardar</button>
        <a href="{{ route('home') }}" class="btn btn-secondary btn-xs">
Cancelar</a>
    </div>
</div>
</form>
<hr>
</div>
</div>
</div>
</div>
</div>
{!!Form::close()!!}
</div>
@endsection

```

Vistas del sistema: en la figura fig.35 se muestra la vista del sistema correspondiente a la historia de usuario HU11.



Fig. 35. Vista de "mi perfil".

3.1.3.2.4 Historia de usuario HU04

Tabla XXI. Historia de usuario "Dashboard".

Historia de usuario	
Código: HU04	Usuario: Encargado de la mesa de ayuda
Nombre de la historia: Dashboard	
Prioridad en la organización: Alta	Riesgo en desarrollo: Media
Iteración asignada: Sprint 02	Dependencia para su desarrollo: Ninguno
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Encargado de la mesa de ayuda. Quiero: Tener un dashboard personalizado según el usuario. Para: tener un cuadro con la información resumida y Controlar y hacer seguimientos de tickets y sus estados.	
Criterios de aceptación: <ul style="list-style-type: none">- El sistema muestra el resumen de la cantidad de tickets por personal de soporte.- El sistema muestra accesos directos con el estado de tickets (pendientes, de hoy, sin asignar, total)	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería mostrar el resumen de la cantidad de tickets por personal de soporte.
- El sistema debería mostrar accesos directos con el estado de tickets (pendientes, de hoy, sin asignar, total)

Prototipos: en la figura fig.36 se muestra el prototipo correspondiente a la historia de usuario HU04.

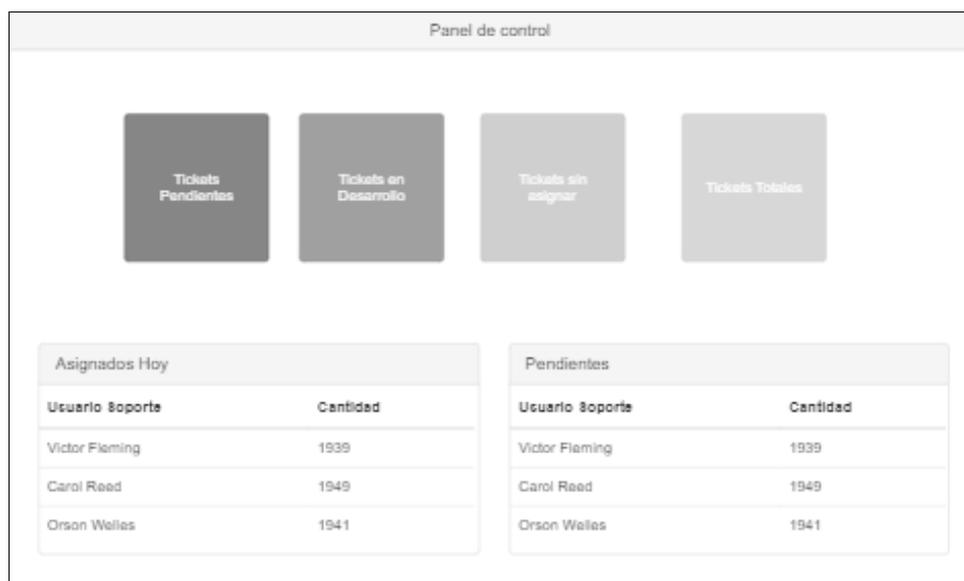


Fig. 36. Prototipo de "panel de control".

Código fuente: a continuación, se muestra parte del código fuente correspondiente a la historia de usuario HU04. Donde se desarrolla el dashboard, para lo cual se crea el controlador “HomeController”.

Código del controlador para el dashboard:

```
<?php

class HomeController extends Controller
{
    public function index()
    {
        $us = Auth::user()->id;
        $mytime = Carbon::now()->locale('es');
        $fecha = $mytime->isoFormat('dddd D MMMM Y');
        $time= $mytime->toDateString();

        //tickets pendientes por usuario
        $tickets_pendientes = DB::table('ticket as t')
        ->join('users as u', 't.user_id', '=', 'u.id')
        ->join('rrhh.tobrero as obr', 'obr.dni', '=', 'u.dni')
        ->select(
            DB::raw('CONCAT(obr.nombre," ",obr.paterno," ",obr.materno) AS usuario'),
            DB::raw('count(*) as pendientes'))
        ->where('t.status_id', '=', '1')
        ->groupBy('usuario')
        ->get();
        //tickets por usuario
        $tickets_usuarios = DB::table('ticket as t')
        ->join('users as u', 't.user_id', '=', 'u.id')
        ->join('rrhh.tobrero as obr', 'obr.dni', '=', 'u.dni')
        ->select(
            DB::raw('CONCAT(obr.nombre," ",obr.paterno," ",obr.materno) AS usuario'),
            DB::raw('count(*) as tickets'))
        ->where('t.created_at', 'LIKE', '%'.$time.'%') //fecha actual
        ->groupBy('usuario')
        ->get();
        //tickets por usuario
        $mis_ticteks = DB::table('ticket')
        ->selectRaw('count(*) as total')
        ->selectRaw("count(case when status_id = '1' then 1 end) as pendiente")
        ->selectRaw("count(case when status_id = '2' then 1 end) as desarrollo")
        ->selectRaw("count(case when created_at LIKE '%$time%' then 1 end) as hoy")
        ->where('ticket.user_id', '=', $us)
        ->first();
    }
}
```

```

        return
    view('dashboard', ["users"=>$tickets_usuarios, "fecha"=>$fecha, "tickets_pendientes"=>$tickets_pendientes, "mistiktes"=>$mis_ticteks]);
    }
}

```

Código de la vista “dashboard”:

```

@section('contenido')
<div class="container">
    <div class="row">
        <div class="col-md-12">
            <div class="card">
                <div class="card-header text-center">
                    <h1><strong>Panel de Control</strong></h1>
                </div>
                <div class="card-body text-center">
                    @if (session('status'))
                        <div class="alert alert-info alert-dismissible">
                            <button type="button" class="close" data-dismiss="alert" aria-label="Close">
                                <span aria-hidden="true">&times;</span>
                            </button>
                            <p>{{ session('status') }}</p>
                        </div>
                    @endif
                    @unless (Auth::user()->esSoporte())
                        <div class="container">
                            <div class="row">
                                <div class="col-md-4 col-xl-3">
                                    <div class="card bg-c-pink order-card">
                                        <div class="card-block">
                                            <h6 class="m-b-20">Tickets</h6>
                                            <h2 class="text-right"><a
href="{{route('ticket.index')}}"><i
class="fa fa-bell f-left text-white"></i></a>
<span>{{tdo_ticket->pendiente}}</span></h2>
                                            <p class="m-b-0">Pendientes</p>
                                        </div>
                                    </div>
                                </div>
                                <div class="col-md-4 col-xl-3">
                                    <div class="card bg-c-blue order-card">
                                        <div class="card-block">
                                            <h6 class="m-b-20">Tickets</h6>
                                            <h2 class="text-right">
                                                <i class="fa fa-hourglass-half f-left text-white"></i>
                                                <span>{{tdo_ticket->hoy}}</span></h2>
                                            <p class="m-b-0">De hoy</p>
                                        </div>
                                    </div>
                                </div>
                            </div>
                    @endunless
                </div>
            </div>
        </div>
    </div>
</div>

```

```

        </div>
    </div>
    <div class="col-md-4 col-xl-3">
        <div class="card bg-c-yellow order-card">
            <div class="card-block">
                <h6 class="m-b-20">Tickets</h6>
                <h2 class="text-right"><a
href="{{route('ticket.sinasignar')}}"><i
                                class="fa fa-refresh f-left text-
white"></i></a><span>{{ $tdo_ticket->sinasignar }}</span></h2>
                <p class="m-b-0">Sin asignar</p>
            </div>
        </div>
    </div>
    <div class="col-md-4 col-xl-3">
        <div class="card bg-c-green order-card">
            <div class="card-block">
                <h6 class="m-b-20">Tickets</h6>
                <h2 class="text-right">
<a href="{{route('ticket.total')}}"><i class="fa fa-database f-left text-
white"></i></a><span>{{ $tdo_ticket->total }}</span></h2>
                <p class="m-b-0">Total</p>
            </div>
        </div>
    </div>
</div>
</div>
@endumless
@if (Auth::user()->esSoporte())
    <div class="container">
        <div class="row">
            <div class="col-md-4 col-xl-3">
                <div class="card bg-c-pink order-card">
                    <div class="card-block">
                        <h6 class="m-b-20">Mis Tickets</h6>
                        <h2 class="text-right"><a href="{{route('mis.tickets')}}"><i
                                class="fa fa-bell f-left text-white"></i></a>
<span>{{ $mistiktes->pendiente }}</span></h2>
                        <p class="m-b-0">Pendientes</p>
                    </div>
                </div>
            </div>
            <div class="col-md-4 col-xl-3">
                <div class="card bg-c-blue order-card">
                    <div class="card-block">
                        <h6 class="m-b-20">Mis Tickets</h6>
                        <h2 class="text-right"><a
href="{{route('mis.desarrollo')}}">

```


Vistas del sistema: en la figura fig.37 se muestra las vistas del sistema correspondiente a la historia de usuario HU04.

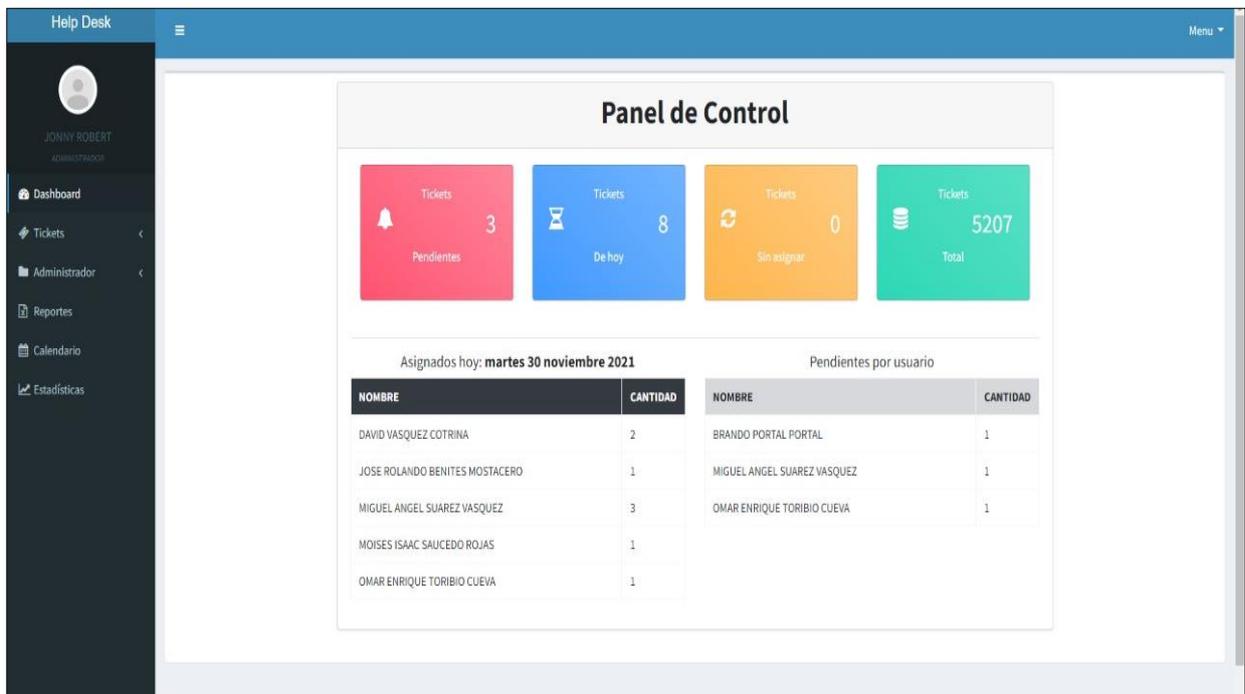


Fig. 37. Vista de dashboard.

3.1.3.3 GRÁFICO BURDOWN

En la siguiente imagen (fig.38) se detalla el proceso de avance del sprint 1, lo cual da como resultado el grafico burdown (fig.39).

Duración del Sprint		15	Esfuerzo Faltante en los días siguientes...																		
Tendencia calculada en los últimos		15	Días	Totales	59	59	54	49	46	42	35	35	30	26	23	19	14	4	2	15	
ID Tarea	Tarea	ID Historia	Responsable	Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
TH6-1	Definir clases a utilizar para gestión de incidentes	HU06	David	Terminado	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH6-2	Definir campos y tipo de información para registrar incidentes	HU06	David	Terminado	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH6-3	Realizar interfaces y formularios	HU06	David	Terminado	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH6-4	Validar interfaces y formularios	HU06	David	Terminado	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH6-5	Realizar diagrama de clases	HU06	David	Terminado	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH6-6	Programar gestión de incidentes	HU06	David	Terminado	6	6	6	6	3	1	0	0	0	0	0	0	0	0	0	0	0
TH7-1	Definir clases a utilizar	HU07	David	Terminado	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0
TH7-2	Definir los accesos cada rol	HU07	David	Terminado	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0	0	0
TH7-3	Diseñar interfaces y formularios	HU07	David	Terminado	3	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0
TH7-4	Realizar diagrama de clases	HU07	David	Terminado	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
TH7-5	Programar gestión de mis tickets	HU07	David	Terminado	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0
TH11-1	Definir clases a utilizar	HU11	David	Terminado	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
TH11-2	Definir tipo de información a consignar en el perfil	HU11	David	Terminado	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
TH11-3	Diseñar interfaces y formularios	HU11	David	Terminado	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0
TH11-4	Realizar diagrama de clases	HU11	David	Terminado	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
TH11-5	Programar administrar perfil	HU11	David	Terminado	6	6	6	6	6	6	6	6	6	6	3	0	0	0	0	0	0
TH4-1	Definir clases a utilizar	HU04	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
TH4-2	Definir tipo de información a consignar en el dashboard	HU04	David	Terminado	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0
TH4-3	Diseñar interfaces y formularios	HU04	David	Terminado	4	4	4	4	4	4	4	4	4	4	4	4	4	4	0	0	0
TH4-4	Realizar diagrama de clases	HU04	David	Terminado	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
TH4-5	Programar dashboard	HU04	David	Terminado	6	6	6	6	6	6	6	6	6	6	6	6	6	3	1	0	0

Fig. 38. Backlog detallado del sprint 2.

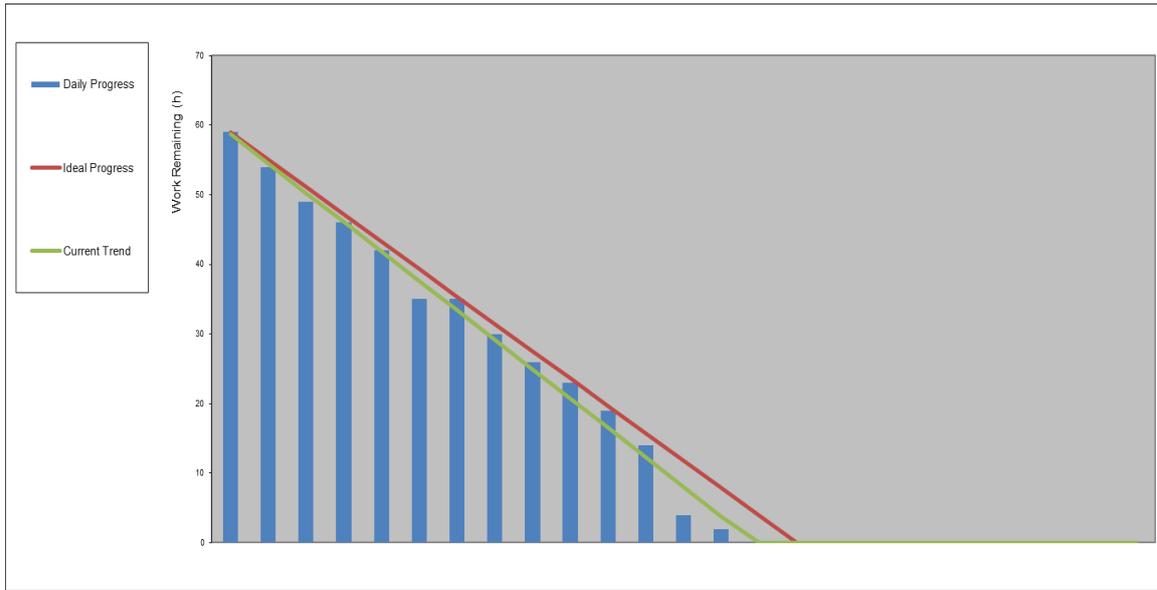


Fig. 39. Gráfico burndown del Sprint 2.

3.1.3.4 REVISIÓN, PRUEBAS Y RETROALIMENTACIÓN DEL SPRINT

Durante esta ceremonia se revisa el incremento, es decir, lo que se realizó durante el Sprint 2, y se analizan los cambios que tuvo el Backlog del producto. También se mostró al dueño del producto el incremento obtenido después de finalizar el Sprint. En la siguiente tabla se detalla los resultados de las pruebas del Sprint.

Tabla XXII. Prueba de funcionalidad Sprint 2.

Historia de Usuario	Criterio de aceptación	Resultado esperado	Resultado obtenido
HU06 Gestión de tickets.	El sistema registra el incidente.	El sistema muestra un mensaje de registro con éxito.	EXITOSO
	El sistema muestra un listado ordenado con los incidentes.	El sistema presenta la lista de los incidentes registrados.	EXITOSO
	El sistema actualiza o edita el ticket del incidente.	El sistema muestra un mensaje de actualizado con éxito.	EXITOSO
	El sistema elimina el ticket de atención.	El sistema muestra un mensaje de eliminado con éxito.	EXITOSO
HU07 Gestión de mis tickets.	El sistema muestra una lista con mis tickets.	El sistema valida el usuario que ha iniciado sesión.	EXITOSO
	El sistema permite cambiar de estado un ticket seleccionado.	Se muestra un desplegable con los estados (pendiente, en desarrollo y atendido)	EXITOSO
	El sistema permite agregar un comentario o descripción de la atención realizada.	Se muestra un campo para ingresar un comentario, y tiene un contador de caracteres.	EXITOSO
HU11 Perfil de usuario.	El sistema muestra mis datos.		EXITOSO
	El sistema permite cambiar contraseña de mi usuario.	La contraseña debe ser mínimo de 6 caracteres.	EXITOSO
	El sistema permite cargar una foto.	Permite cargar archivo tipo JPG y PNG.	EXITOSO
HU04 Dashboard	El sistema muestra el resumen de la cantidad de tickets por personal de soporte.	<ul style="list-style-type: none"> - Se muestra tickets pendientes, en desarrollo, tickets de hoy y total. - Cada cuadro es un acceso directo a ver dicha información en más detalle. 	EXITOSO

3.1.4 SPRINT 03

3.1.4.1 SPRINT BACKLOG

Según el producto backlog o pila del producto, se estableció el tercer sprint (Sprint 3) con las historias de usuario HU05, HU09 y HU10 (de la tabla XXIV a la tabla XXVI), los cuales sirvieron para realizar el backlog del sprint (Tabla XXIII).

Tabla XXIII. Backlog de sprint 3.

Historia de Usuario		Tareas	
ID	Deseo...	ID	Descripción
HU05	Tener un calendario y poder agregar, editar y eliminar un evento en una fecha seleccionada.	TH5-1	Definir clases a utilizar
		TH5-2	Definir tipo de información a consignar en el calendario
		TH5-3	Diseñar interfaces y formularios de calendario
		TH5-4	Validar interfaces y formularios
		TH5-5	Realizar diagrama de clases
		TH5-6	Programar calendario
HU09	Tener un apartado donde se muestre estadísticas acerca de la atención de incidentes y la productividad del equipo de soporte.	TH9-1	Definir clases a utilizar
		TH9-2	Definir tipo de información a consignar en estadísticas
		TH9-3	Diseñar interfaces de estadísticas
		TH9-4	Realizar diagrama de clases
		TH9-5	Programar estadísticas
HU10	Generar el reporte en Excel con las atenciones de los incidentes registrados.	TH10-1	Definir clases a utilizar
		TH10-2	Definir tipo de información a consignar en el reporte
		TH10-3	Diseñar interfaces de reporte
		TH10-4	Realizar diagrama de clases
		TH10-5	Programar reporte

3.1.4.2 DESARROLLO DEL SPRINT

3.1.4.2.1 Historia de usuario HU05

Tabla XXIV. Historia de usuario "gestión de eventos".

Historia de usuario	
Código: HU05	Usuario: Supervisor
Nombre de la historia: Gestión de eventos.	
Prioridad en la organización: Alta	Riesgo en desarrollo: Baja
Iteración asignada: Sprint 03	Dependencia para su desarrollo: Ninguno
Programador responsable: David Vásquez Cotrina	
Descripción: Como: Encargado de la mesa de ayuda. Quiero: Tener un calendario y poder agregar, editar y eliminar un evento en una fecha seleccionada. Para: anotar atenciones futuras y tomar notas.	
Criterios de aceptación: <ul style="list-style-type: none">- El sistema agrega un evento en una fecha seleccionada.- El sistema edita el evento registrado.- El sistema elimina el evento registrado.	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería agregar un evento en una fecha seleccionada.
- El sistema debería editar el evento.
- El sistema debería eliminar el evento.

Prototipos: en la figura fig.40 se muestra el prototipo correspondiente a la historia de usuario HU05.

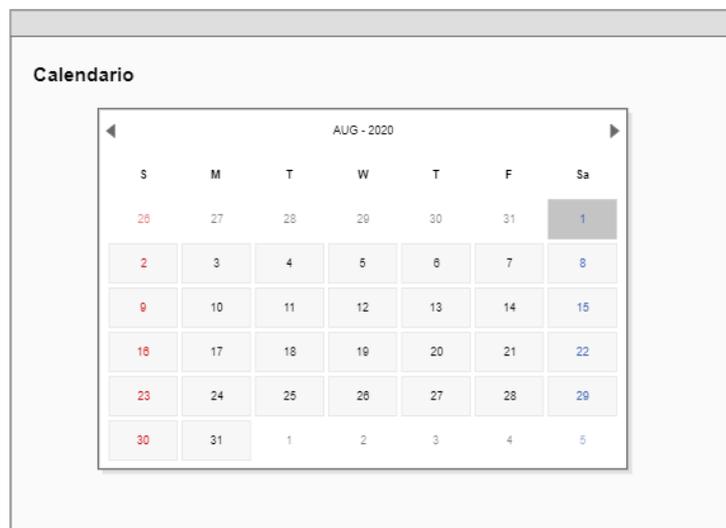


Fig. 40. Prototipo "calendario".

Diagrama de clases: en la figura fig.41 se muestra el diagrama de clases correspondientes a la historia de usuario HU05.

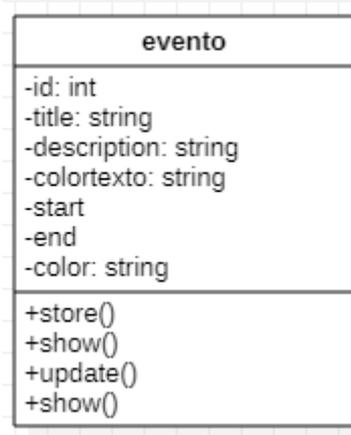


Fig. 41. Diagrama de clases de "evento".

Código fuente: a continuación, se muestra parte del código fuente correspondiente a la historia de usuario HU05. Para lo cual se creó el modelo “evento” y su controlador.

Código del modelo evento:

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;

class Evento extends Model
{
    protected $table = 'eventos';
    protected $primaryKey = 'id';
    public $timestamps=false;
    protected $fillable = ['title', 'description', 'color', 'colortext', 'start', 'end'];
    protected $guarded = [];
}
```

Código del controlador para calendario:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Evento;
use Validator;

class CalendarioController extends Controller
{
    public function __construct()
    {
```

```

        $this->middleware('auth');
    }
    public function index()
    {
        return view('calendar.index');
    }
    public function store(Request $request)
    {
        $datosEvento = request()->except(['_token', '_method']);
        Evento::insert($datosEvento);
    }
    public function show()
    {
        $data['eventos'] = Evento::all();
        return response()->json($data['eventos']);
    }

    public function destroy($id)
    {
        $eventos = Evento::findOrFail($id);
        Evento::destroy($id);
        return response()->json($id);
    }
    public function update(Request $request,$id)
    {
        $datosEvento = request()->except(['_token', '_method']);
        $respuesta = Evento::where('id', '=', $id)->update($datosEvento);
        return response()->json($respuesta);
    }
}

```

Código de vista calendario:

```

@extends('layouts.admin')
@section('contenido')
<div class="container-fluid">
    <div class="row">
        <div class="col-2">
        </div>
        <div class="col-8">
            <div class="card">
                <div id='calendar'></div>
            </div> <!--end card -->
        </div>
    </div>
    <br>
</div>
<!-- Modal -->

```

```

<div class="modal fade" id="exampleModal" tabindex="-1" role="dialog" aria-
labelledby="exampleModallabel"
  aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModallabel">Datos del Evento</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <div class="d-none">
          <!-- hide-->
          <input type="text" name="txtID" id="txtID" class="form-control">
          fecha:
          <input type="text" name="txtFecha" id="txtFecha" class="form-control">
        </div>

        <div class="form-row">
          <div class="alert-danger" id="alerta"></div>
          <div class="form-group col-md-8">
            <label for="">Título:</label>
            <input type="text" name="txtTitulo" id="txtTitulo" class="form-control" required>
          </div>
          <div class="form-group col-md-4">
            <label for="">Hora:</label>
            <input type="time" min="07:00" max="19:00" step="600" name="txtHora" id="txtHora"
class="form-control" required>
          </div>
          <div class="form-group col-md-12">
            <label for="">Descripción: <em>(400 caracteres máx.)</em></label>
            <textarea name="txtDescripcion" id="txtDescripcion" cols="30" rows="4" class="form-
control" required maxlength="400"></textarea>
            <em><p id="totalc"></p></em>
          </div>
          <div class="form-group col-md-12">
            <label for="">Color:</label>
            <input type="color" name="txtColor" id="txtColor" class="form-control">
          </div>
        </div>
      </div>
      <div class="modal-footer">
        <button id="btnAgregar" class="btn btn-success">Agregar</button>
        <button id="btnEliminar" class="btn btn-danger">Eliminar</button>
        <button id="btnModificar" class="btn btn-warning">Modificar</button>
        <button id="btnCancelar" class="btn btn-secondary" data-dismiss="modal">Cancelar</button>
      </div>
    </div>
  </div>
</div>

```

```

    </div>
</div>
@endsection
@push('scripts')
<script>
    url_="{{url('/calendario')}}";
    url_show="{{ url('/calendario/show')}}";
</script>
<script src="{{asset('js/myjs/event.js')}}"></script>
<script>
    $('#txtDescripcion').on("input", function(){
        var maxLength = $(this).attr("maxlength");
        var currentLength = $(this).val().length;

        if(currentLength >= maxLength){
            alertify.error('Has alcanzado el límite de caracteres');
        }else{
            $('#totalc').html(maxLength - currentLength+" restantes");
        }
    });
</script>
@endpush

```

Vistas del sistema: en la figura fig.42 se muestra las vistas del sistema correspondiente a la historia de usuario HU05.



Fig. 42. Vista de calendario.

3.1.4.2.2 Historia de usuario HU09

Tabla XXV. Historia de usuario "estadísticas".

Historia de usuario	
Código: HU09	Usuario: Supervisor
Nombre de la historia: Estadísticas	
Prioridad en la organización: Alta	Riesgo en desarrollo: Baja
Iteración asignada: Sprint 03	Dependencia para su desarrollo: Ninguno
Programador responsable: David Vásquez Cotrina	
Descripción:	
<p>Como: Supervisor.</p> <p>Quiero: Tener un apartado donde se muestre estadísticas acerca de la atención de incidentes y la productividad del equipo de soporte. Por ejemplo: cantidad de tickets según la categoría del incidente, áreas u oficinas que solicitan más atenciones, y cantidad de tickets registrados por mes.</p> <p>Para: tener información útil para la toma de decisiones.</p>	
Criterios de aceptación:	
<ul style="list-style-type: none"> - El sistema muestra un gráfico estadístico de cantidad de tickets por personal de soporte. - Se muestra grafico tickets por categoría. - Se muestra áreas u oficinas con más tickets. - Se muestra la tendencia de registro de tickets según meses del año. 	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema muestra un gráfico estadístico de cantidad de tickets por personal de soporte.
- Se muestra grafico tickets por categoría.
- Se muestra áreas u oficinas con más tickets.
- Se muestra la tendencia de registro de tickets según meses del año.

Prototipos: en la figura fig.41 se muestra el prototipo correspondiente a la historia de usuario HU09.

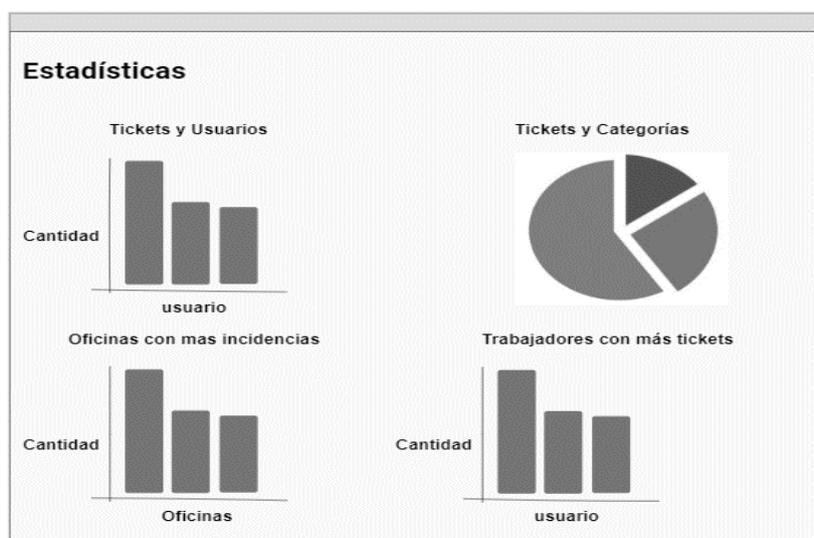


Fig. 43. Prototipo de vista estadísticas.

Diagrama de clases: en la figura fig.44 se muestra el diagrama de clases correspondientes a la historia de usuario HU09.

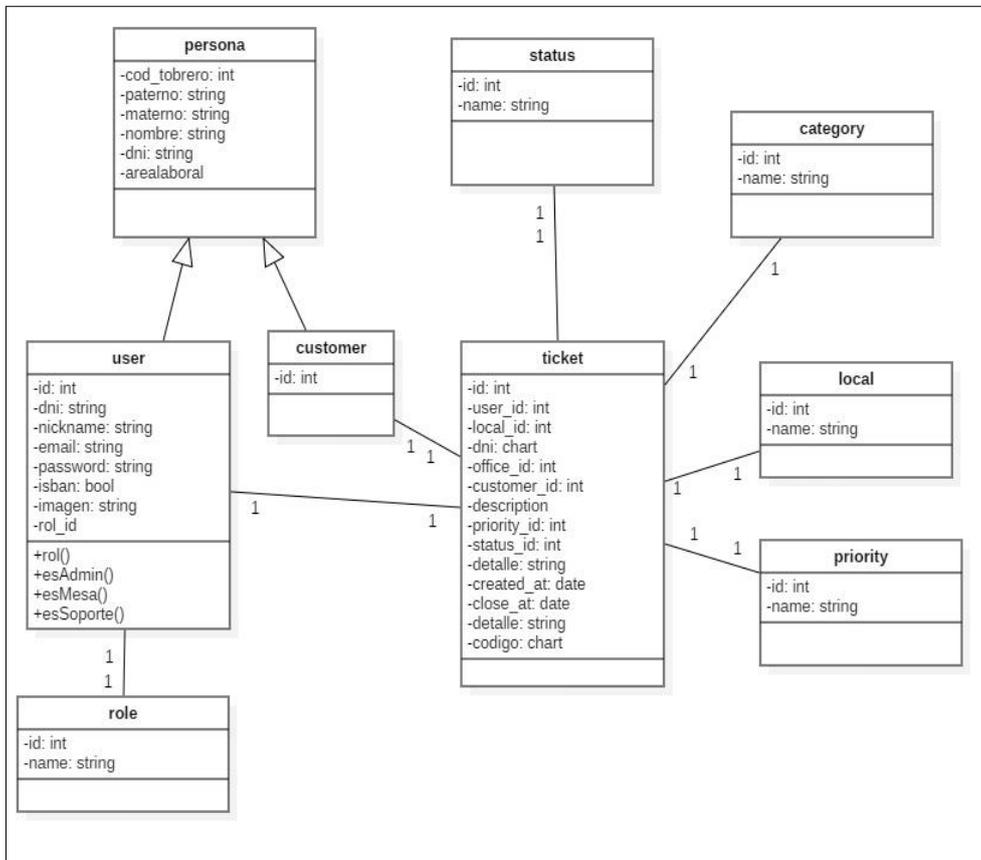


Fig. 44. Diagrama de clases de la HU09

Código fuente: a continuación, se muestra parte del código fuente correspondiente a la historia de usuario HU09. Donde se agregó los métodos `milistartool()`, `datos()` y `datosmensual()` en el controlador “ConsultaController”.

Código del controlador “ConsultaController”:

```

<?php
namespace App\Http\Controllers;
use Validator;
use App\Ticket;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class ConsultaController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
}
  
```

```

}
public function datos(Request $request)
{
    $tickets_usuarios = DB::table('ticket as t')
    ->join('users as u', 't.user_id', '=', 'u.id')
    ->join('rrhh.tobrero as obr', 'obr.dni', '=', 'u.dni')
    ->select('obr.nombre as usuario', DB::raw('count(*) as tickets'))
    ->groupBy('obr.nombre')
    ->orderBy('tickets', 'desc')
    ->get();
    $tickets_areas = DB::table('ticket as t')
    ->join('rrhh.x_areas as ar', 'ar.id', '=', 't.office_id')
    ->select('ar.nombre as area', DB::raw('count(*) as tickets'))
    ->groupBy('ar.nombre')
    ->orderBy('tickets', 'desc')
    ->take(5)
    ->get();
    $tickets_categoria = DB::table('ticket as t')
    ->join('category as ca', 'ca.id', '=', 't.category_id')
    ->select('ca.name as categoria', DB::raw('count(*) as tickets'))
    ->groupBy('ca.name')
    ->orderBy('tickets', 'asc')
    ->get();
    $tickets_trabajador = DB::table('ticket as t')
    ->join('rrhh.tobrero as obr', 't.dni', '=', 'obr.dni')
    ->select(
        DB::raw('CONCAT(obr.nombre, " ", obr.paterno, " ", obr.materno) AS trabajador'),
        DB::raw('count(*) as tickets'))
    ->groupBy('trabajador')
    ->orderBy('tickets', 'desc')
    ->take(5)
    ->get();
    $anio = Carbon::now()->year;
    $tickets_fecha = DB::table('ticket as t')
    ->select(
        DB::raw("MONTH(t.created_at) as month"),
        DB::raw('count(*) as tickets'))
    ->where('t.created_at', 'LIKE', '%'.$anio.'%')
    ->orderBy('month', 'asc')
    ->groupBy('month')
    ->get();
    foreach ($tickets_fecha as $ticket) {
        switch ($ticket->month) {
            case 1:
                $ticket->mes = "Enero";
                break;
            case 2:
                $ticket->mes = "Febrero";
                break;

```

```

        case 3:
            $ticket->mes = "Marzo";
            break;
        case 4:
            $ticket->mes = "Abril";
            break;
        case 5:
            $ticket->mes = "Mayo";
            break;
        case 6:
            $ticket->mes = "Junio";
            break;
        case 7:
            $ticket->mes = "Julio";
            break;
        case 8:
            $ticket->mes = "Agosto";
            break;
        case 9:
            $ticket->mes = "Septiembre";
            break;
        case 10:
            $ticket->mes = "Octubre";
            break;
        case 11:
            $ticket->mes = "Noviembre";
            break;
        case 12:
            $ticket->mes = "Diciembre";
            break;
    }
}
return response()->json(['datos'=>$tickets_usuarios,
                        'datos1'=>$tickets_areas,
                        'datos2'=>$tickets_categoria,
                        'datos3'=>$tickets_trabajador,
                        'datos4'=>$tickets_fecha
                        ]);
}
public function datosMensual($anio)
{
    $tickets_fecha = DB::table('ticket as t')
->select(
        DB::raw("MONTH(t.created_at) as month"),
        DB::raw('count(*) as tickets'))
->where('t.created_at', 'LIKE', '%'.$anio.'%')
->orderBy('month', 'asc')
->groupBy('month')
->get();
}

```

```

foreach ($tickets_fecha as $ticket) {
    switch ($ticket->month) {
        case 1:
            $ticket->mes = "Enero";
            break;
        case 2:
            $ticket->mes = "Febrero";
            break;
        case 3:
            $ticket->mes = "Marzo";
            break;
        case 4:
            $ticket->mes = "Abril";
            break;
        case 5:
            $ticket->mes = "Mayo";
            break;
        case 6:
            $ticket->mes = "Junio";
            break;
        case 7:
            $ticket->mes = "Julio";
            break;
        case 8:
            $ticket->mes = "Agosto";
            break;
        case 9:
            $ticket->mes = "Septiembre";
            break;
        case 10:
            $ticket->mes = "Octubre";
            break;
        case 11:
            $ticket->mes = "Noviembre";
            break;
        case 12:
            $ticket->mes = "Diciembre";
            break;
    }
}
$total_row = $tickets_fecha->count();
if ($total_row == 0) {
    return response()->json(['error'=>'No hay datos para mostrar']);
} else {
    return response()->json(["dato" => $tickets_fecha]);
}
}
}

```

Código de la vista de estadísticas:

```
@section('contenido')
<div class="container">
  <div class="card-header">
    <h2><strong>Estadísticas</strong> </h2>
  </div>
  <div class="card-body">
    <div class="row">
      <div class="col-lg-6 col-md-12 col-sm-12 col-xs-12">
        <label for="" class="form-control text-center">Tickets y Usuarios</label>
        <canvas id="myChart" width="100%" height="80"></canvas>
      </div>
      <div class="col-lg-6 col-md-12 col-sm-12 col-xs-12">
        <label for="" class="form-control text-center">Tickets y Categorías</label>
        <canvas id="myChart2" width="100%" height="80"></canvas>
      </div>
    </div>
    <hr>
    <div class="row">
      <div class="col-lg-6 col-md-12 col-sm-12 col-xs-12">
        <label for="" class="form-control text-center">Oficinas con mayor cantidad de
incidencias</label>
        <canvas id="myChart1" width="100%" height="80"></canvas>
      </div>
      <div class="col-lg-6 col-md-12 col-sm-12 col-xs-12">
        <label for="" class="form-control text-center">Trabajadores con más tickets
registrados</label>
        <canvas id="myChart3" width="100%" height="80"></canvas>
      </div>
    </div>
    <hr>
    <div class="row">
      <label for="" class="form-control text-center">Tendencia de incidencias por mes</label>
      <div class="col-4">
        <label>Año: </label>
        <select name="fecha" id="fecha">
          <option value="0">Seleccionar</option>
          @for ($i = $anio_inicio; $i < $anio+1; $i++)
            <option value="{{ $i }}">{{ $i }}</option>
          @endfor
        </select>
      </div>
      <div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
        <canvas id="myChart4" width="100%" height="40"></canvas>
      </div>
    </div>
    <div class="row text-center" id="mensual">
```

```

<div class="col-lg-12 col-md-12 col-sm-12 col-xs-12">
  <label for="" class="form-control text-center">Incidencias mensuales</label>
  <canvas id="myChart4" width="100%" height="40"></canvas>
  <table class="table table-bordered table-condesed" id="dataTable">
    <thead>
      <th>Mes</th>
      <th>Cantidad</th>
    </thead>
    <tbody id="tbody">
    </tbody>
  </table>
</div>
</div>
</div>
<!--fin div row -->
@push('scripts')
<script>
  var meses=[];
var valores4=[];
var usuarios=[];
var valores=[];
var areas=[];
var valores1=[];
var categorias=[];
var valores2=[];
var trabajadores=[];
var valores3=[];
var colores = [];

$('#fecha').on('change',function() {
  meses=[];
  valores4=[];
  myChartfav.destroy();
  $('#tbody').empty();
  codigo = $(this).val();
$.ajax({
  url:"{{ url('datos_mensual') }}" + codigo,
  success:function(resp){
    if (resp.error) {
      alertify.error(resp.error);
      var todo='<tr><td>'+resp.error+'</td></tr>'
      $('#tbody').append(todo);
    }
    else{
      for (let x = 0; x < resp.dato.length; x++) {
        var todo='<tr><td>'+resp.dato[x].mes+'</td>'
        todo+='<td>'+resp.dato[x].tickets+'</td></tr>'
        $('#tbody').append(todo);
      }
    }
  }
});

```

```

        meses.push(resp.dato[x].mes);
        valores4.push(resp.dato[x].tickets);
    }
    generarGrafica4();
}
}
});
});
$(document).ready(function() {
    $.ajax({
        url: '/mostrar_datos',
        method: 'POST',
        data: {
            id:1,
            _token: $("meta[name='csrf-token']").attr("content")
        }
    }).done(function(resp){
        for (let x = 0; x < resp.datos.length; x++)
        {
            usuarios.push(resp.datos[x].usuario);
            valores.push(resp.datos[x].tickets);
        }
        generarGrafica();

        for (let x = 0; x < resp.datos1.length; x++) {
            areas.push(resp.datos1[x].area);
            valores1.push(resp.datos1[x].tickets);
        }
        generarGrafica1();

        for (let x = 0; x < resp.datos2.length; x++) {
            categorias.push(resp.datos2[x].categoria);
            valores2.push(resp.datos2[x].tickets);

            r = Math.floor(Math.random() * 200);
            g = Math.floor(Math.random() * 200);
            b = Math.floor(Math.random() * 200);
            o = '0.7';
            color = 'rgb(' + r + ', ' + g + ', ' + b + ', ' + o + ')';
            colores.push(color);
        }
        generarGrafica2();

        for (let x = 0; x < resp.datos3.length; x++) {
            trabajadores.push(resp.datos3[x].trabajador);
            valores3.push(resp.datos3[x].tickets);
        }
        generarGrafica3();
    });
});

```

```

        for (let x = 0; x < resp.datos4.length; x++) {
            meses.push(resp.datos4[x].mes);
            valores4.push(resp.datos4[x].tickets);
        }
        generarGrafica4();
    })
});

function getRandomColor() {
    var letters = '0123456789ABCDEF'.split('');
    var color = '#';
    for (var i = 0; i < 6; i++) {
        color += letters[Math.floor(Math.random() * 16)];
    }
    return color;
}

function generarGrafica(params) {
    var ctx = document.getElementById('myChart').getContext('2d');
    var myChart = new Chart(ctx, {
        type: 'bar',
        data: {
            labels: usuarios,
            datasets: [{
                label: 'Tickets',
                data: valores,
                backgroundColor: '#33AEEF',
                borderWidth: 0
            }]
        },
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    },
                    scaleLabel: {
                        display: true,
                        labelString: "Cantidad"
                    }
                }]
            }
        }
    });
}

function generarGrafica1(params) {
    var ctx = document.getElementById('myChart1').getContext('2d');
    var myDoughnutChart = new Chart(ctx, {
        type: 'horizontalBar',
        data: {

```

```

        labels: areas,
        datasets: [{
            label: 'Tickets',
            data: valores1,
            backgroundColor: "#f1c40f",
            borderWidth: 0
        }]
    },
    options: {
        scales: {
            yAxes: [{
                ticks: {
                    beginAtZero: true
                }
            }]
        }
    }
});
}

function generarGrafica2(params) {
    var ctx = document.getElementById('myChart2').getContext('2d');
    var myPieChart = new Chart(ctx, {
        type: 'pie',
        data: {
            labels: categorias,
            datasets: [{
                data: valores2,
                backgroundColor: colores,
                borderWidth: 1
            }]
        },
        options: {
            legend: {
                display: true
            },
            animation: {
                duration: 1200,
                easing: "easeOutQuart",
                onComplete: function () {
                    var ctx = this.chart.ctx;
                    ctx.font='14px LatoRegular, Helvetica,sans-serif';
                    ctx.textAlign = 'center';
                    ctx.textBaseline = 'bottom';
                    this.data.datasets.forEach(function (dataset) {
                        for (var i = 0; i < dataset.data.length; i++) {
                            var m = dataset._meta[Object.keys(dataset._meta)[0]].data[i]._model,
                                t = dataset._meta[Object.keys(dataset._meta)[0]].total,
                                mR = m.innerRadius + (m.outerRadius - m.innerRadius) / 2,
                                sA = m.startAngle,

```

```

        eA = m.endAngle,
        mA = sA + (eA - sA)/2;
    var x = mR * Math.cos(mA);
    var y = mR * Math.sin(mA);
    ctx.fillStyle = '#fff';

    var p = String(Math.round(dataset.data[i]/t*100)) + "%";
    if(dataset.data[i] > 0) {
        ctx.fillText(dataset.data[i], m.x + x, m.y + y-10);
        ctx.fillText(p, m.x + x, m.y + y + 5);
    }
}
});
}
}
});
}
function generarGrafica3(params) {
    var ctx = document.getElementById('myChart3').getContext('2d');
    var myChart = new Chart(ctx, {
        type: 'bar',
        data: {
            labels: trabajadores,
            datasets: [{
                label: 'Tickets',
                data: valores3,
                backgroundColor: "#2ecc71",
                borderWidth: 1
            }]
        },
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true,
                    },
                    scaleLabel: {
                        display: true,
                        labelString: "Cantidad"
                    }
                }]
            }
        }
    });
}
var myChartfav;
function generarGrafica4(params) {
    var ctx = document.getElementById('myChart4').getContext('2d');
    myChartfav = new Chart(ctx, {
        type: 'line',

```

```

data: {
  labels: meses,
  datasets: [{
    label: 'Tickets',
    data: valores4,
    backgroundColor: 'rgba(2, 117, 216, 0.2)',
    borderColor: 'rgba(2, 117, 216, 1)',
    borderWidth: 1,
    borderRadius: 5,
    pointBackgroundColor: 'rgba(2,117,216,1)',
    pointBorderColor: 'rgba(255,255,255,0.8)',
    pointHoverRadius: 5,
    pointHoverBackgroundColor: 'rgba(2,117,216,1)',
    pointHitRadius:20,
    pointBorderWidth:2
  }]
},
options: {
  scales: {
    xAxes: [{
      time: {
        unit: 'date'
      },
      gridLines:{
        display:false
      },
      ticks:{
        maxTicksLimit:12
      }
    ]},
    yAxes: [{
      ticks:{
        min:10,
        max:500,
        maxTicksLimit:12,
      },
      gridLines:{
        color: 'rgba(0,0,0,.125)'
      },
      scaleLabel: {
        display: true,
        labelString: "Cantidad"
      }
    ]},
  }
}
}); }

```

```

</script>
@endpush
@endsection

```

Vistas del sistema: en la figura fig.45 se muestra lo vista del sistema correspondiente a la historia de usuario HU09.

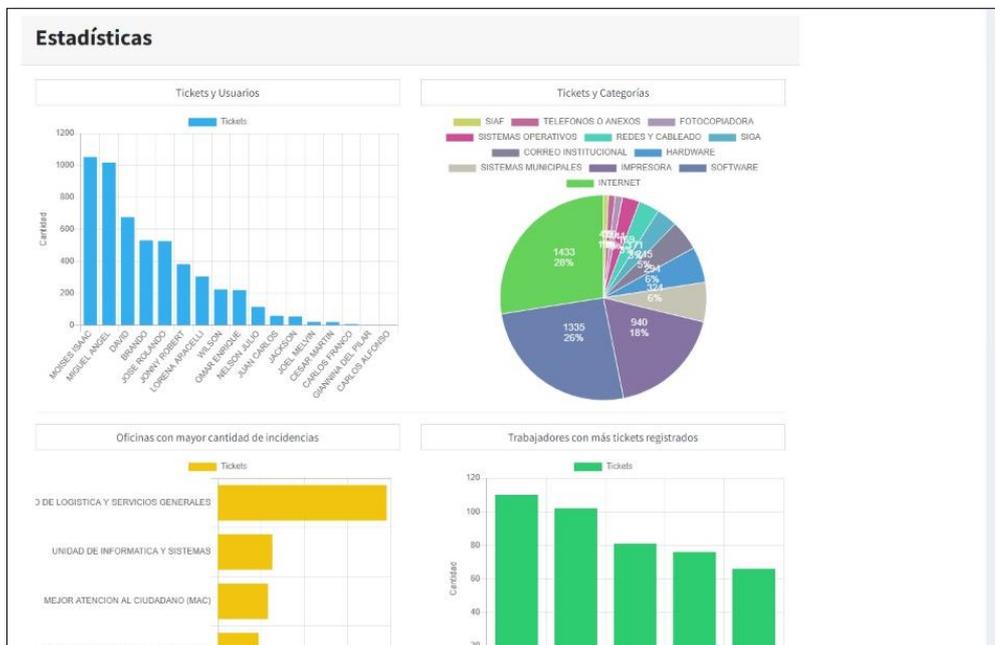


Fig. 45. Vista de "estadísticas".

3.1.4.2.3 Historia de usuario HU10

Tabla XXVI. Historia de usuario "reportes".

Historia de usuario	
Código: HU10	Usuario: Encargado de la mesa de ayuda
Nombre de la historia: Reportes	
Prioridad en la organización: Alta	Riesgo en desarrollo: Baja
Iteración asignada: Sprint 03	Dependencia para su desarrollo: Ninguno
Programador responsable: David Vásquez Cotrina	
Descripción:	
Como: Encargado de la mesa de ayuda.	
Quiero: Generar el reporte en Excel con las atenciones de los incidentes registrados.	
Para: presentar informes.	
Criterios de aceptación:	
- El sistema permite seleccionar fechas	
- El sistema genera un reporte en Excel según las fechas seleccionadas	

De la presente historia de usuario se extraen los requisitos para esta parte del sistema:

- El sistema debería permitir seleccionar fechas.
- El sistema debería generar un reporte en Excel según las fechas seleccionadas

Prototipos: en la figura fig.46 se muestra el prototipo correspondiente a la historia de usuario HU10.

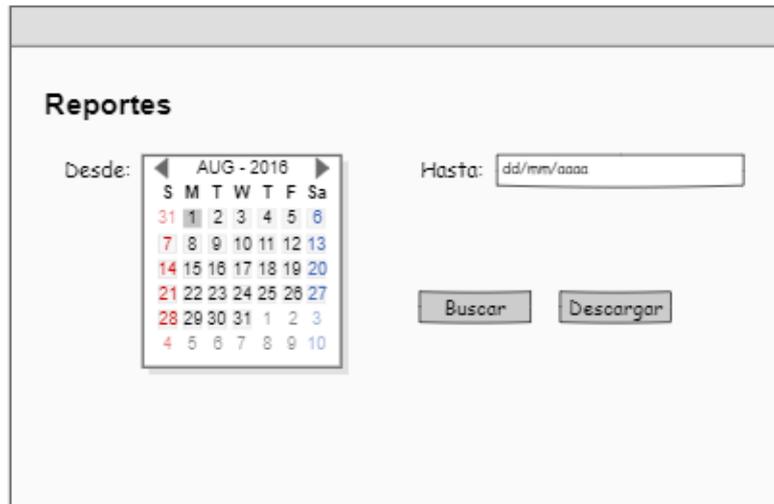


Fig. 46. Prototipo de vista de calendario.

Código fuente: a continuación, se muestra parte del código fuente correspondiente a la historia de usuario HU10.

Código del controlador “ReportesController”:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Exports\TicketExport;
use Illuminate\Support\Facades\DB;
use Maatwebsite\Excel\Facades\Excel;
use Illuminate\Support\Facades\Redirect;

class ReportesController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }
    public function index(Request $request)
    {
        $method = $request->method();

        if ($request->isMethod('post')) {

            $query = trim($request->get('from'));
            $query1 = trim($request->get('to'));
```

```

if ($request->has('search')){

$categorias = DB::table('ticket as t')
->join('rrhh.x_areas as ar','t.office_id','=','ar.id')
->join('rrhh.tobrero as ob','t.customer_id','=','ob.cod_tobrero')
->join('users as u','t.user_id','=','u.id')
->join('local as lo','t.local_id','=','lo.id')
->join('status as s','t.status_id','=','s.id')
->join('priority as p','t.priority_id','=','p.id')
->join('category as ca','t.category_id','=','ca.id')
->join('rrhh.tobrero as obr','obr.dni','=','u.dni')
->select(
    't.id',
    't.codigo as cod',
    't.created_at as fecha',
    't.description',
    'ob.nombre as cliente',
    'ob.paterno as cliente_ap',
    'ob.materno as cliente_am',
    'ar.nombre as oficina',
    'lo.name as local',
    'obr.nombre as user',
    't.detalle as detalle',
    's.name as condi')
->whereBetween('t.created_at', [$query, $query1])
->get();

return view('almacen.reportes.index', ["reportes" => $categorias]);
}

if ($request->has('exportExcel')) {
return Excel::download(new TicketExport($query, $query1),
'Reporte_Sistema_HelDesk.xlsx');
}
}
else
{
$ViewsPage = DB::table('ticket as t')
->join('rrhh.x_areas as ar','t.office_id','=','ar.id')
->join('rrhh.tobrero as ob','t.customer_id','=','ob.cod_tobrero')
->join('users as u','t.user_id','=','u.id')
->join('local as lo','t.local_id','=','lo.id')
->join('status as s','t.status_id','=','s.id')
->join('priority as p','t.priority_id','=','p.id')
->join('category as ca','t.category_id','=','ca.id')
->select('t.id',
    'u.nickname as user',
    'ob.nombre as cliente',

```

```

        'ob.nombre as cliente_ap',
        'ob.materno as cliente_am',
        'ar.nombre as oficina',
        't.created_at as fecha',
        't.description',
        'lo.name as local',
        't.detalle as detalle')
->where('lo.name', '=', 'ninguno')
->get();
return view('almacen.reportes.index', ['reportes'=> $ViewsPage]);
    }
}
}

```

Código de la vista "reportes":

```

@section('titulo')
<div class="col-lg-12 col-md-10 col-sm-10 col-xs-12">
<h3>REPORTES</h3>
</div>
@endsection
@section('contenido')
    <form action="{{url('almacen/reportes')}}" method="POST" enctype="multipart/form-data">
        @csrf
        <div class="container">
            <div class="row">
                <div class="col-lg-4 col-md-5 col-sm-6 col-xs-12">
                    <label for="from" class="col-form-label">Desde:</label>
                    <input type="date" class="form-control input-sm" id="from" name="from" required>
                </div>
                <div class="col-lg-6 col-md-8 col-sm-6 col-xs-12">
                    <br>
                    <button type="submit" class="btn btn-primary btn-sm"
name="search">Buscar</button>
                    <button type="submit" class="btn btn-success btn-sm" name="exportExcel">Exportar
a Excel</button>
                </div>
            </div>
        </div>
    </form>
    <br>
    <div class="table-responsive">
        <table class="table table-bordered table-condesed" id="tblticket">
            <thead class="bg-info">
                <th>#</th>
                <th>Fecha</th>
                <th>Incidente</th>
                <th>Solicitante</th>
                <th>Oficina</th>
            </thead>
        </table>
    </div>

```

```

        <th>Local</th>
        <th>Responsable</th>
        <th>Detalle</th>
        <th>Condición</th>
    </thead>
    @foreach ($reportes as $ViewsPages)
    <tr>
        <td>{{ $ViewsPages->id.'-'. $ViewsPages->cod}}</td>
        <td>{{ $ViewsPages->fecha }}</td>
        <td>{{ $ViewsPages->description }}</td>
        <td>{{ $ViewsPages->cliente.' ' . $ViewsPages->cliente_ap.' ' . $ViewsPages->cliente_am}}</td>
        <td>{{ $ViewsPages->oficina }}</td>
        <td>{{ $ViewsPages->local }}</td>
        <td>{{ $ViewsPages->user}}</td>
        <td>{{ $ViewsPages->detalle }}</td>
        <td>{{ $ViewsPages->condi }}</td>
    </tr>
    @endforeach
</table>
</div>
@endsection
@push('scripts')
<script>
$(document).ready(function() {
    $('#tblticket').DataTable({
        "order": [[ 0, "desc" ]],
        "language":{
            "sProcessing": "Procesando...",
            "sLengthMenu": "Mostrar _MENU_ registros",
            "sZeroRecords": "No hay coincidencias",
            "sEmptyTable": "Ningún dato disponible",
            "info": "Mostrando del _START_ al _END_ de un total de _TOTAL_ registros",
            "sInfoEmpty": "",
            "sInfoFiltered": "",
            "sInfoPostFix": "",
            "sSearch": "Buscar:",
            "sUrl": "",
            "sInfoThousands": ",",
            "sLoadingRecords": "Cargando...",
            "oPaginate": {
                "sFirst": "Primero",
                "sLast": "Último",
                "sNext": "Siguiete",
                "sPrevious": "Anterior"},
            "oAria": {
                "sSortAscending": " : Activar para ordenar la columna de manera ascendente",

```

```

        "sSortDescending": ": Activar para ordenar la columna de manera
descendente"},
        "buttons":{
            "copy": "Copiar",
            "colvis": "Visibilidad"}
    }
});
});
</script>
@endpush

```

Vistas del sistema: en las figuras fig.47 y fig.48 se muestran la vista del sistema correspondiente a la historia de usuario HU10.

REPORTES

Desde: Hasta:

Mostrar registros Buscar:

#	Fecha	Incidente	Solicitante	Oficina	Local	Responsable	Detalle	Condición
Ningún dato disponible								

Fig. 47. Vista "reportes".

REPORTES

Desde: Hasta:

Mostrar registros Buscar:

#	Fecha	Incidente	Solicitante	Oficina	Local	Responsable	Detalle	Condición
9983- YMCC	2022-08-08 14:43:32	usuario solicita atencion con su pc, se colgo	ELMER IVAN ZAVALETA HONORIO	MEJOR ATENCION AL CIUDADANO (MAC)	QHAPAQ ÑAN	BRANDO	se elimino temporales	ATENDIDO
9982- OYD1	2022-08-08 14:39:44	usuario solicita activación de office	SEGUNDO LEONCIO GODOY HUAMAN	UNIDAD DE PLANIFICACIÓN Y DESARROLLO DE PERSONAS	QHAPAQ ÑAN	MOISES ISAAC	Activación de Office	ATENDIDO
9981-IXNF	2022-08-08 14:38:54	usuario solicita atencion para ingresar a su pc	JAIME OVIDIO CUZCO MINCHAN	SUBGERENCIA DE ESTUDIOS Y PROYECTOS	QHAPAQ ÑAN	JOSE ROLANDO	Se quitó del dominio y se volvió a unir para evitar el error de confianza	ATENDIDO
9980- KJQD	2022-08-08 14:23:08	Revisar impresora	EDITH SOLEDAD RAMIREZ DIAZ	UNIDAD DE LOGISTICA Y SERVICIOS GENERALES	QHAPAQ ÑAN	DAVID	Se revisó impresora	ATENDIDO

Fig. 48. Resultado de consultar un reporte.

3.1.4.3 GRÁFICO BURDOWN

En la siguiente imagen (fig.49) se detalla el proceso de avance del sprint 1, lo cual da como resultado el grafico burdown(fig.50).

Duración del Sprint		15	Esfuerzo Faltante en los días siguientes...																		
Tendencia calculada en los últimos		15	Días	Totales	56	56	51	47	44	41	39	36	32	28	25	19	13	8	2		
ID Tarea	Tarea	ID Historia	Responsable	Estado	Est.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
TH5-1	Definir clases a utilizar	HU05	David	Terminado	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH5-2	Definir tipo de información a consignar en el calendario	HU05	David	Terminado	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH5-3	Diseñar interfaces y formularios de calendario	HU05	David	Terminado	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH5-4	Validar interfaces y formularios	HU05	David	Terminado	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TH5-5	Realizar diagrama de clases	HU05	David	Terminado	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0
TH5-6	Programar calendario	HU05	David	Terminado	8	8	8	8	8	5	3	0	0	0	0	0	0	0	0	0	0
TH9-1	Definir clases a utilizar	HU09	David	Terminado	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
TH9-2	Definir tipo de información a consignar en estadísticas	HU09	David	Terminado	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	0
TH9-3	Diseñar interfaces de estadísticas	HU09	David	Terminado	4	4	4	4	4	4	4	4	4	0	0	0	0	0	0	0	0
TH9-4	Realizar diagrama de clases	HU09	David	Terminado	3	3	3	3	3	3	3	3	3	3	0	0	0	0	0	0	0
TH9-5	Programar estadísticas	HU09	David	Terminado	8	8	8	8	8	8	8	8	8	8	8	2	0	0	0	0	0
TH10-1	Definir clases a utilizar	HU10	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
TH10-2	Definir tipo de información a consignar en el reporte	HU10	David	Terminado	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0
TH10-3	Diseñar interfaces de reporte	HU10	David	Terminado	3	3	3	3	3	3	3	3	3	3	3	3	3	0	0	0	0
TH10-4	Realizar diagrama de clases	HU10	David	Terminado	3	3	3	3	3	3	3	3	3	3	3	3	3	1	0	0	0
TH10-5	Programar reporte	HU10	David	Terminado	7	7	7	7	7	7	7	7	7	7	7	7	7	2	0	0	0

Fig. 49. Backlog detallado del sprint 3.

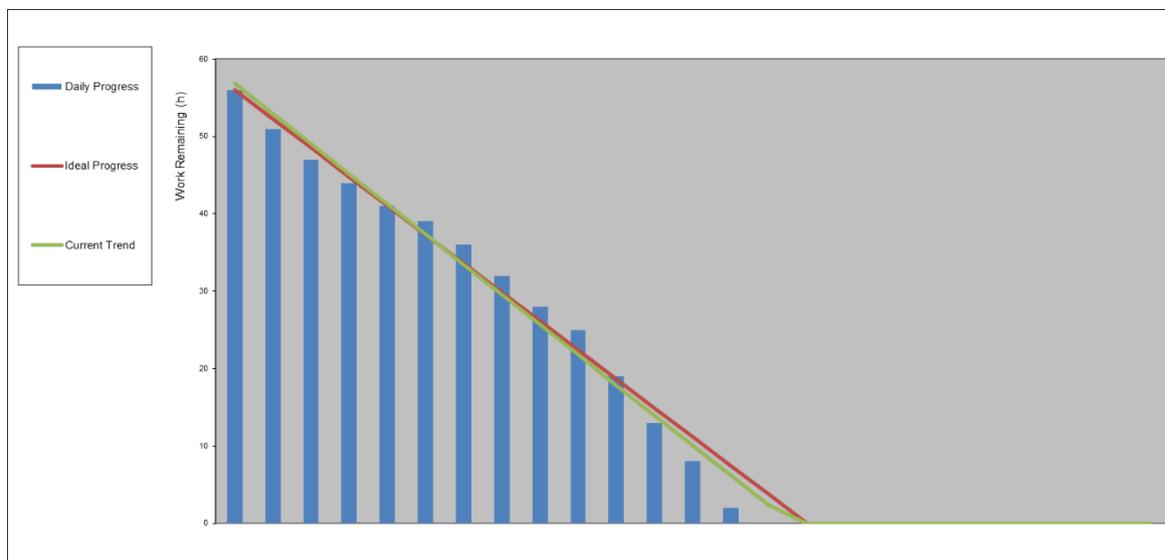


Fig. 50. Gráfico burdown del Sprint 3.

3.1.4.4 REVISIÓN, PRUEBAS Y RETROALIMENTACIÓN DEL SPRINT

Durante esta ceremonia se revisa el incremento, es decir, lo que se realizó durante el Sprint 3, y se analizan los cambios que tuvo el Backlog del producto. También se mostró al dueño del producto el incremento obtenido después de finalizar el Sprint. En la siguiente tabla se detalla los resultados de las pruebas del Sprint. Dado que éste es el último Sprint, los avances se han unido a los anteriores y se ha procedido a poner en producción el sistema, tal como se muestran en las imágenes del Anexo 9.

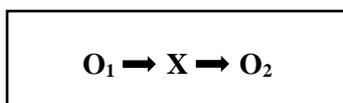
Tabla XXVII. Prueba de funcionalidad sprint 3.

Historia de Usuario	Criterio de aceptación	Resultado esperado	Resultado obtenido
HU05 Gestión de eventos.	El sistema agrega un evento en una fecha seleccionada.	Al seleccionar una fecha, se muestra un formulario flotante.	EXITOSO
	El sistema edita el evento registrado.	El sistema muestra un mensaje de actualizado con éxito.	EXITOSO
	El sistema elimina el evento registrado.	El sistema muestra un mensaje de eliminado con éxito.	EXITOSO
HU09 Estadísticas.	El sistema muestra un gráfico estadístico de cantidad de tickets por personal de soporte.	El sistema presenta un gráfico ordenado.	EXITOSO
	Se muestra gráfico tickets por categoría.	El sistema presenta un gráfico ordenado.	EXITOSO
	Se muestra áreas u oficinas con más tickets.	El sistema presenta un gráfico ordenado.	EXITOSO
	Se muestra la tendencia de registro de tickets según meses del año.	El sistema presenta un gráfico ordenado.	EXITOSO
HU10 Reportes.	El sistema permite seleccionar fechas.	El sistema muestra un calendario flotante para seleccionar la fecha.	EXITOSO
	El sistema genera un reporte en Excel según las fechas seleccionadas.	Se descarga el archivo en formato exe.	EXITOSO

3.2 TRATAMIENTO Y ANÁLISIS DE DATOS Y PRESENTACIÓN DE RESULTADOS

3.2.1 TRATAMIENTO

La presente investigación es de **tipo** aplicada, porque se busca la aplicación o utilización de conocimientos, teorías y principios en una situación real. Además, es de **nivel** descriptiva, puesto que se describen los datos y características de la población en estudio. En cuanto al **diseño**, se usa el diseño experimental, dentro del cual se utiliza el Pre-Experimental, dado que se realiza una prueba previa (pre-test) al estímulo o tratamiento experimental, y una prueba posterior (post-test) al estímulo o tratamiento experimental.



Donde:

X: Tratamiento

O₁: Medición pre - test

O₂: Medición post - test

3.2.1.1 VARIABLES E INDICADORES

En la siguiente tabla se presenta las variables de estudio y los indicadores asumidos para la presente investigación.

Tabla XXVIII. Matriz de variables, dimensiones e indicadores de la investigación.

Variable	Dimensión	Indicadores
Sistema Help Desk	Usabilidad	Grado de operabilidad o capacidad para ser usado.
		Grado de accesibilidad.
		Grado de aprendizaje.
Gestión de incidentes	Tiempo	Tiempo empleado para registrar un incidente
		Tiempo empleado para asignar un incidente.
		Tiempo empleado para la elaboración de reportes.
	Satisfacción del equipo de soporte.	Nivel de satisfacción con respecto al proceso de gestión de incidentes.
		Nivel de satisfacción con respecto a los tiempos del proceso gestión de incidentes.
	Productividad	Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo.

3.2.1.2 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

En cuanto a la recolección de los datos, se ha empleado diferentes técnicas, tales como la *observación*, que mediante su instrumento **ficha de observación** permitió recabar y registrar información sobre la dimensión tiempo. Por consiguiente, se aplicó la técnica *encuesta* y su instrumento **cuestionario**, lo que posibilitó obtener datos para las dimensiones usabilidad y satisfacción de los usuarios del sistema. Finalmente, para medir la dimensión productividad se empleó la técnica **análisis documental** con el instrumento ficha de análisis documental. Todos estos instrumentos se muestran en el [anexo 1](#).

Cabe señalar que la medición de la dimensión usabilidad, perteneciente a la variable independiente, se realizó una sola vez, luego de haberse implementado el Sistema Help Desk. No obstante, la medición para las dimensiones tiempo, satisfacción de los usuarios del sistema y productividad se efectuaron en dos oportunidades, un antes (pre-test) y un después (post-test) de la implementación de la herramienta.

3.2.1.3 VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Los instrumentos de recolección de datos adoptados en la presente investigación son: cuestionario, ficha de observación y ficha de análisis documental, lo cuales están detallados en el [anexo 1](#). Asimismo, cada instrumento ha sido validado por expertos, según se muestra en el [anexo 3](#). Además, para evaluar la confiabilidad de los algunos instrumentos se ha utilizado el coeficiente Alpha de Cronbach, como se aprecia en el [anexo 2](#).

3.2.2 ANÁLISIS DE DATOS

Para el análisis de los datos, se ha dividido las variables de estudio en dimensiones, tal como se describe en la [tabla XXVIII](#), en ese sentido se analiza cada dimensión con sus respectivos indicadores:

3.2.2.1 DIMENSIÓN TIEMPO

Para el análisis de esta dimensión, se procedió a observar y registrar los tiempos (en minutos) de las actividades pertenecientes al proceso actual de atención de incidentes (Fig.51 y Fig.52), con la finalidad de obtener tiempos base, los cuales sirven para determinar el número de observaciones a realizar en la aplicación del instrumento, para ello se utilizó la tabla de **General Electric** ([Anexo 4](#)), en ese sentido, los datos obtenidos se muestran en la tabla XXIX. Cabe

precisar que el instrumento ficha de observación fue aplicado en dos momentos (pre-test y post-test), así mismo, el post-test se aplicó al nuevo proceso de gestión de incidentes (ver [anexo 9](#)).

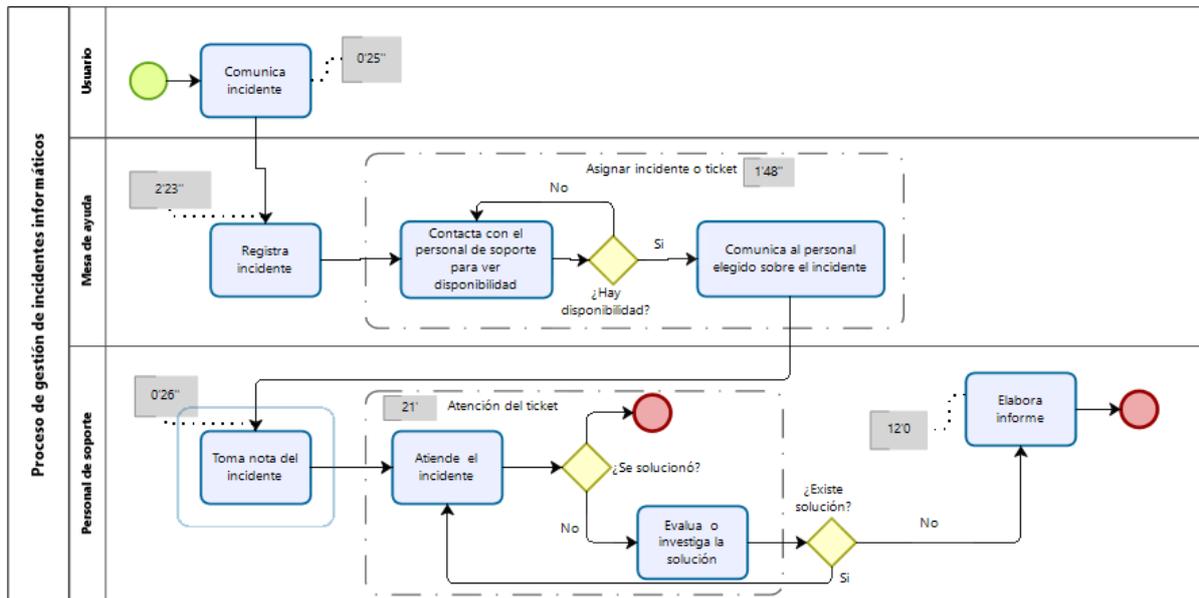


Fig. 51. Proceso inicial de Gestión de Incidentes.

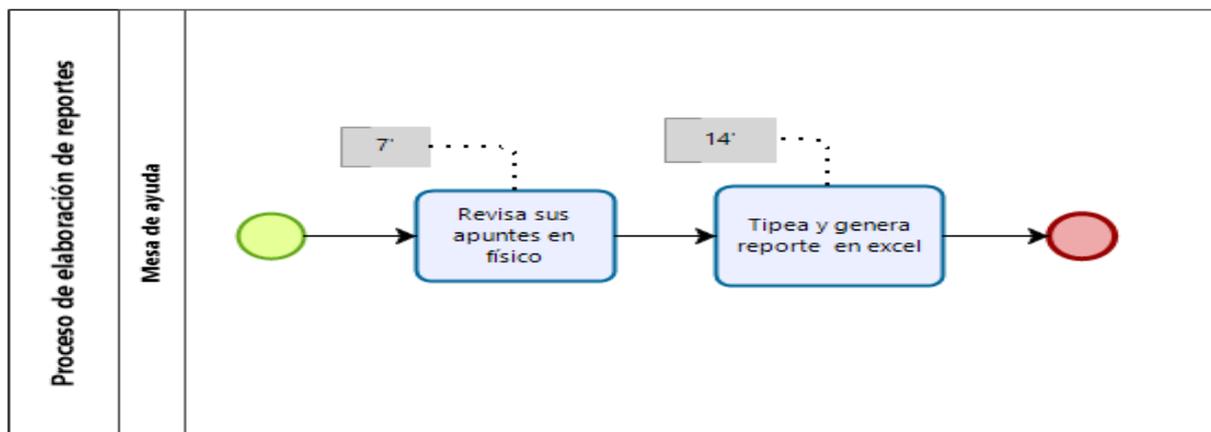


Fig. 52. Proceso inicial de Elaboración de reportes.

A continuación, se presenta una tabla con el resumen de los datos obtenidos de la toma de tiempos a los procesos iniciales, así como también se muestra el número de observaciones a realizar en la aplicación del instrumento “ficha de observación” para medir la dimensión tiempo.

Tabla XXIX. Dimensión tiempo, indicadores, tiempos y número de observaciones.

Indicador	Tiempo estimado (min)	Número de observaciones a realizar
Tiempo empleado para registrar un incidente	2.38	20
Tiempo empleado para asignar un incidente	1.8	25
Tiempo empleado para la elaboración de reportes.	21	5

Una vez determinado el número de observación a realizar, se procedió con la recolección de datos para cada indicador. Además, para contrastar las hipótesis se empleó la prueba T-Student para muestras pareadas, la cual es aplicable a muestras menores o iguales a treinta (≤ 30) y que los datos provengan de una distribución normal [35]. En ese sentido, en el [anexo 3](#) se ofrece a detalle los resultados de pruebas de normalidad.

3.2.2.1.1 Indicador: “Tiempo empleado para registrar un incidente”.

Se realizaron 20 observaciones para la obtención de datos del indicador tiempo empleado para registrar un incidente (Ver [anexo 5](#)); en la siguiente tabla se muestra el resumen de los resultados conseguidos:

Tabla XXX. Resumen de observaciones de tiempo empleado para registrar un incidente.

TIEMPO EMPLEADO PARA REGISTRAR UN INCIDENTE (MINUTOS)		
CICLOS	TAS	TDS
C1	2.50	1.77
C2	0.98	1.08
C3	1.65	1.42
C4	1.45	1.25
C5	2.60	0.42
C6	2.17	0.65
C7	2.67	0.07
C8	2.17	0.63
C9	2.50	0.68
C10	1.83	0.63
C11	1.53	0.85
C12	1.65	0.65
C13	1.35	0.62
C14	1.20	0.65
C15	1.28	0.60
C16	1.42	0.67
C17	2.67	0.68
C18	2.52	0.53
C19	1.93	0.43
C20	1.97	0.28

Donde las variables son:

TAS: Tiempo empleado para registrar un incidente antes del uso del Sistema.

TDS: Tiempo empleado para registrar un incidente después del uso del Sistema.

Prueba de hipótesis: Aplicando la prueba **T-Student**, se establece la hipótesis nula, la hipótesis alternativa, nivel de significancia y nivel de confianza:

Hipótesis nula: *el tiempo empleado para registrar un incidente después del uso del sistema es mayor o igual que el tiempo empleado para registrar un incidente antes del uso del sistema.*

$$H_0 = TDS \geq TAS$$

Hipótesis alternativa: *el tiempo empleado para registrar un incidente después del uso del sistema es menor que el tiempo empleado para registrar un incidente antes del uso del sistema.*

$$H_a = TDS < TAS$$

Nivel de significancia: *el nivel de significancia es del 5%.*

$$\alpha = 0.05$$

Nivel de confianza: *el nivel de confianza es del 95%.*

$$1-\alpha = 0.95$$

Donde, además, se cumple que si:

P-valor \leq Alpha, Se rechaza H_0 y se acepta la H_a

P-valor \geq Alpha, Se rechaza H_a y se acepta la H_0

Luego de aplicar la prueba T-Student y procesar los datos recolectados con el software SPSS, los resultados obtenidos son como se detallan en la siguiente imagen:

		Prueba de muestras emparejadas							
		Diferencias emparejadas							
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	TAS - TDS	1,269	,691	,154	,946	1,592	8,216	19	,000

Fig. 53. Prueba T-Student para tiempo de registro de un incidente.

El p-valor (0.000) obtenido es menor o está por debajo del nivel de significancia (0.05), Por lo tanto, se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa:

H_a : *el tiempo empleado para registrar un incidente después del uso del sistema es menor que el tiempo empleado para registrar un incidente antes del uso del sistema.*

3.2.2.1.2 Indicador: “Tiempo empleado para asignar un incidente”.

Una vez determinado el número de observación a realizar, se procedió a la recolección de datos para el indicador; se realizaron 25 observaciones para el proceso de asignar un incidente (Ver [anexo 5](#)). En la siguiente tabla se muestra el resumen de los resultados obtenidos:

Tabla XXXI. Resumen de observaciones de tiempo empleado para asignar un incidente

TIEMPO EMPLEADO PARA ASIGNAR UN INCIDENTE (MINUTOS)		
CICLOS	TAS	TDS
C1	1.62	0.15
C2	1.48	0.13
C3	1.67	0.17
C4	0.95	0.13
C5	0.67	0.18
C6	1.32	0.17
C7	1.24	0.20
C8	0.97	0.12
C9	1.17	0.10
C10	0.97	0.13
C11	0.98	0.17
C12	1.28	0.15
C13	1.43	0.18
C14	1.17	0.15
C15	1.54	0.12
C16	1.12	0.20
C17	0.72	0.10
C18	0.92	0.08
C19	1.51	0.17
C20	1.72	0.13
C21	1.15	0.14
C22	1.65	0.14
C23	1.90	0.11
C24	0.95	0.08
C25	1.27	0.10

Donde las variables son:

TAS: Tiempo empleado para asignar un incidente antes del uso del Sistema.

TDS: Tiempo empleado para asignar un incidente después del uso del Sistema.

Prueba de hipótesis: Aplicando la prueba **T-Student**, se establece la hipótesis nula, la hipótesis alternativa, nivel de significancia y nivel de confianza:

Hipótesis nula: *el tiempo empleado para asignar un incidente después del uso del sistema es mayor o igual que el tiempo empleado para asignar un incidente antes del uso del sistema.*

$$H_0 = TDS \geq TAS$$

Hipótesis alternativa: *el tiempo empleado para asignar un incidente después del uso del sistema es menor que el tiempo empleado para asignar un incidente antes del uso del sistema.*

$$H_a = TDS < TAS$$

Nivel de significancia: *el nivel de significancia es del 5%.*

$$\alpha = 0.05$$

Nivel de confianza: *el nivel de confianza es del 95%.*

$$1-\alpha = 0.95$$

Donde, además, se cumple que si:

P-valor \leq Alpha, Se rechaza H_0 y se acepta la H_a

P-valor \geq Alpha, Se rechaza H_a y se acepta la H_0

Luego de realizar la prueba T-Student y procesar los datos recolectados con el software SPSS, los resultados obtenidos se muestran en las siguientes imágenes:

Prueba de muestras emparejadas									
Diferencias emparejadas									
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	TAS - TDS	1,11600	,32057	,06411	,98367	1,24833	17,406	24	,000

Fig. 54. Prueba T-Student para tiempo para asignar un incidente.

Donde el p-valor (0.000) obtenido es menor o está por debajo del nivel de significancia (0.05), Entonces dado ese resultado, se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_a):

H_a : *el tiempo empleado para asignar un incidente después del uso del sistema es menor que el tiempo empleado para asignar un incidente antes del uso del sistema.*

3.2.2.1.3 Indicador: “Tiempo empleado para la elaboración de reportes”.

Una vez determinado el número de observación a realizar, se procedió a la recolección de datos para el indicador; se realizaron 20 observaciones para el proceso de elaboración de reportes (Ver [anexo 5](#)). En la siguiente tabla se muestra el resumen de los resultados obtenidos:

Tabla XXXII. Resumen de observaciones de tiempo empleado para la elaboración de reportes.

TIEMPO EMPLEADO PARA LA ELABORACIÓN DE REPORTES (MINUTOS)		
CICLOS	TAS	TDS
C1	8.65	0.27
C2	10.9	0.28
C3	9.48	0.25
C4	8.45	0.30
C5	9.08	0.20

Donde las variables son:

TAS: Tiempo empleado para la elaboración de reportes antes del uso del sistema.

TDS: Tiempo empleado para la elaboración de reportes después del uso del sistema.

Prueba de hipótesis: Aplicando la prueba **T-Student**, se establece la hipótesis nula, la hipótesis alternativa, nivel de significancia y nivel de confianza:

Hipótesis nula: *el tiempo empleado para la elaboración de reportes después del uso del sistema es mayor o igual que el tiempo empleado para la elaboración de reportes antes del uso del sistema.*

$$H_0 = TDS \geq TAS$$

Hipótesis alternativa: *el tiempo empleado para la elaboración de reportes después del uso del sistema es menor que el tiempo empleado para la elaboración de reportes antes del uso del sistema.*

$$H_a = TDS < TAS$$

Nivel de significancia: *el nivel de significancia es del 5%.*

$$\alpha = 0.05$$

Nivel de confianza: el nivel de confianza es del 95%.

$$1-\alpha = 0.95$$

Donde, además, se cumple que si:

P-valor \leq Alpha, Se rechaza Ho y se acepta la Ha

P-valor \geq Alpha, Se rechaza Ha y se acepta la Ho

Luego de realizar la prueba T-Student y procesar los datos recolectados con el software SPSS, los resultados obtenidos se muestran en las siguientes imágenes:

Prueba de muestras emparejadas									
Diferencias emparejadas									
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	TAS - TDS	11,83400	,83173	,37196	10,80127	12,86673	31,815	4	,000

Fig. 55. Prueba T-Student para tiempo para la elaboración de reportes.

Donde el p-valor (0.000) obtenido es menor o está por debajo del nivel de significancia (0.05), Entonces dado ese resultado, se rechaza la hipótesis nula (Ho) y se acepta la hipótesis alternativa (**Ha**):

Ha: el tiempo empleado para la elaboración de reportes después del uso del sistema es menor que el tiempo empleado para la elaboración de reportes antes del uso del sistema.

3.2.2.2 DIMENSIÓN SATISFACCIÓN DEL EQUIPO SOPORTE

Para el análisis de esta dimensión, se aplicó una encuesta con escala de Likert (ver [anexo 1](#)) a los trabajadores de la Unidad de Informática y Sistemas involucrados en la gestión de incidentes, con la finalidad de obtener datos para los indicadores propuestos. Este instrumento fue aplicado en dos ocasiones (ver [anexo 6](#)), antes del uso del sistema (pre-test) y después del uso del sistema (post-test). Además, para contrastar las hipótesis se empleó la prueba T-Student para muestras pareadas.

3.2.2.2.1 Indicador: “Nivel de satisfacción con respecto al proceso de gestión de incidentes”.

Tabla XXXIII. Resumen de nivel de satisfacción respecto al proceso de gestión de incidentes

NIVEL DE SATISFACCIÓN CON RESPECTO AL PROCESO DE GESTIÓN DE INCIDENTES.		
ITEM	NSA	NSD
PRE_1	2	4,4
PRE_2	2	3,4
PRE_3	2,3	3,5
PRE_4	1,8	4,2
PRE_5	2,8	4,5

Donde las variables son:

NSA: Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes antes del uso del sistema.

NSD: Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes después del uso del sistema.

Prueba de hipótesis: Se aplicó la prueba **T-Student**, la cual establece la hipótesis nula, la hipótesis alternativa, nivel de significancia y nivel de confianza:

Hipótesis nula: *Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes antes del uso del sistema es mayor o igual que el Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes después del uso del sistema.*

$$H_0 = NSA \geq NSD$$

Hipótesis alternativa: Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes después del uso del sistema es mayor que el Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes antes del uso del sistema.

$$H\alpha = NDS > NSA$$

Nivel de significancia: el nivel de significancia es del 5%.

$$\alpha = 0.05$$

Nivel de confianza: el nivel de confianza es del 95%.

$$1-\alpha = 0.95$$

Donde, además, se cumple que si:

P-valor \leq Alpha, Se rechaza Ho y se acepta la H α

P-valor \geq Alpha, Se rechaza H α y se acepta la Ho

Luego de realizar la prueba T-Student y procesar los datos recolectados en el software SPSS, se obtiene el siguiente resultado:

Prueba de muestras emparejadas									
Diferencias emparejadas									
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	NSA - NSD	-3,08000	,67231	,30067	-3,91478	-2,24522	-10,244	4	,001

Fig. 56. Prueba t-student nivel de satisfacción respecto al proceso de gestión de incidentes.

Donde el p-valor (0.001) obtenido es menor o está por debajo del nivel de significancia (0.05), Entonces dado ese resultado, se rechaza la hipótesis nula (Ho) y se acepta la hipótesis alternativa (**H α**):

H α : Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes después del uso del sistema es mayor que el Nivel de satisfacción del equipo de soporte con respecto al proceso de gestión de incidentes antes del uso del sistema.

3.2.2.2.2 Indicador: “Nivel de satisfacción con respecto a los tiempos del proceso de gestión de incidentes”.

Tabla XXXIV. Resumen de nivel de satisfacción respecto al proceso de gestión de incidentes

NIVEL DE SATISFACCIÓN CON RESPECTO A LOS TIEMPOS DEL PROCESO DE GESTIÓN DE INCIDENTES.		
ITEM	NSTA	NSTD
PRE_1	1,5	4,6
PRE_2	1,3	4,5
PRE_3	1,2	4,7
PRE_4	1,0	5,0
PRE_5	1,0	5,0

Donde las variables son:

NSTA: Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes antes del uso del sistema.

NSTD: Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes después del uso del sistema.

Prueba de hipótesis: Se aplicó la prueba **T-Student**, la cual establece la hipótesis nula, la hipótesis alternativa, nivel de significancia y nivel de confianza:

Hipótesis nula: *Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes antes del uso del sistema es mayor o igual que el Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes después del uso del sistema.*

$$H_0 = NSTA \geq NSTD$$

Hipótesis alternativa: *Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes después del uso del sistema es mayor que el Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes antes del uso del sistema.*

$$H_a = NSTD > NSTA$$

Nivel de significancia: *el nivel de significancia es del 5%.*

$$\alpha = 0.05$$

Nivel de confianza: el nivel de confianza es del 95%.

$$1-\alpha = 0.95$$

Donde, además, se cumple que si:

P-valor \leq Alpha, Se rechaza Ho y se acepta la H α

P-valor \geq Alpha, Se rechaza H α y se acepta la Ho

Luego de realizar la prueba T-Student y procesar los datos recolectados en el software SPSS, se obtiene el siguiente resultado:

Prueba de muestras emparejadas									
Diferencias emparejadas									
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	NSTA - NSTD	-3,56000	,42778	,19131	-4,09117	-3,02883	-18,608	4	,000

Fig. 57. Prueba t-student nivel de satisfacción respecto a los tiempos del proceso de gestión de incidentes.

Donde el p-valor (0.000) obtenido es menor o está por debajo del nivel de significancia (0.05), Entonces dado ese resultado, se rechaza la hipótesis nula (Ho) y se acepta la hipótesis alternativa (**H α**):

H α : Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes después del uso del sistema es mayor que el Nivel de satisfacción del equipo de soporte con respecto a los tiempos del proceso de gestión de incidentes antes del uso del sistema.

3.2.2.3 DIMENSIÓN PRODUCTIVIDAD

Para el análisis de esta dimensión, se consideró el indicador “número de incidentes atendidos persona del equipo de soporte durante un día de trabajo”, por lo que se aplicó la técnica de análisis documental con su instrumento ficha de análisis documental (ver [anexo 1](#)). En donde se revisaron los archivos y documentos utilizados en la Unidad de Informática y Sistemas para llevar un control sobre los incidentes reportados. En el [anexo 7](#) se puede revisar a detalle la recolección de datos del instrumento ficha de análisis documental. Por consiguiente, se aplicó la prueba T-stundet para analizar los datos.

3.2.2.3.1 Indicador: “número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo”.

Tabla XXXV. Resumen de productividad del equipo de soporte.

NÚMERO DE INCIDENTES ATENDIDOS POR PERSONA DEL EQUIPO DE SOPORTE DURANTE UN DÍA DE TRABAJO		
Trabajador	PAS	PDS
TR_1	3,6	4
TR_2	2,6	4
TR_3	3,8	4
TR_4	3,4	4,4
TR_5	3,6	3,2
TR_6	2,6	2,8
TR_7	2,4	4,2
TR_8	1,2	1,2
TR_9	2,2	3,8
TR_10	1,4	1,6

Donde las variables son:

PAS: Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo antes del uso del sistema.

PDS: Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo después del uso del sistema.

Prueba de hipótesis: Se aplicó la prueba **T-Student**, la cual establece la hipótesis nula, la hipótesis alternativa, nivel de significancia y nivel de confianza:

Hipótesis nula: *Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo antes del uso del sistema es mayor que el Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo después del uso del sistema.*

$$H_0 = PDS < PAS$$

Hipótesis alternativa: *Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo después del uso del sistema es mayor que el Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo antes del uso del sistema.*

$$H_a = PDS > PAS$$

Nivel de significancia: *el nivel de significancia es del 5%.*

$$\alpha = 0.05$$

Nivel de confianza: *el nivel de confianza es del 95%.*

$$1-\alpha = 0.95$$

Donde, además, se cumple que si:

P-valor \leq Alpha, Se rechaza H_0 y se acepta la H_a

P-valor \geq Alpha, Se rechaza H_a y se acepta la H_0

Luego de realizar la prueba T-Student y procesar los datos recolectados en el software SPSS, se obtiene el siguiente resultado:

Prueba de muestras emparejadas									
Diferencias emparejadas									
		Media	Desv. Desviación	Desv. Error promedio	95% de intervalo de confianza de la diferencia		t	gl	Sig. (bilateral)
					Inferior	Superior			
Par 1	PAS - PDS	-,64000	,75307	,23814	-1,17871	-,10129	-2,687	9	,025

Fig. 58. Prueba t-student nivel de satisfacción respecto a los tiempos del proceso de gestión de incidentes.

Donde el p-valor (0.025) obtenido es menor o está por debajo del nivel de significancia (0.05), Entonces dado ese resultado, se rechaza la hipótesis nula (H_0) y se acepta la hipótesis alternativa (H_a):

***H_a :** Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo después del uso del sistema es mayor que el Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo antes del uso del sistema.*

3.2.3 PRESENTACIÓN DE RESULTADOS

Una vez realizado el tratamiento y análisis de los datos respecto a las variables de estudio, se presenta los resultados haciendo uso de tablas y gráficos estadísticos, tal como se detalla a continuación:

3.2.3.1 USABILIDAD

En cuanto a la dimensión Usabilidad, se aplicó una encuesta basada en la escala de Likert a los trabajadores que conforman el equipo de soporte. Esta aplicación tuvo lugar posterior a la implementación de la herramienta Sistema Help Ddesk, con la finalidad de obtener datos para medir el grado de usabilidad del sistema. Los datos recolectados se muestran en el [anexo 8](#) y los resultados obtenidos después de procesarlos se muestran en los siguientes gráficos:

Grado de operabilidad.

Respecto a este punto, el 51% de los encuestados consideran que están totalmente de acuerdo, a su vez, el 47% manifiesta que está de acuerdo y el 2% de los mismos indica estar ni de acuerdo ni en desacuerdo. Así lo muestra el siguiente gráfico:

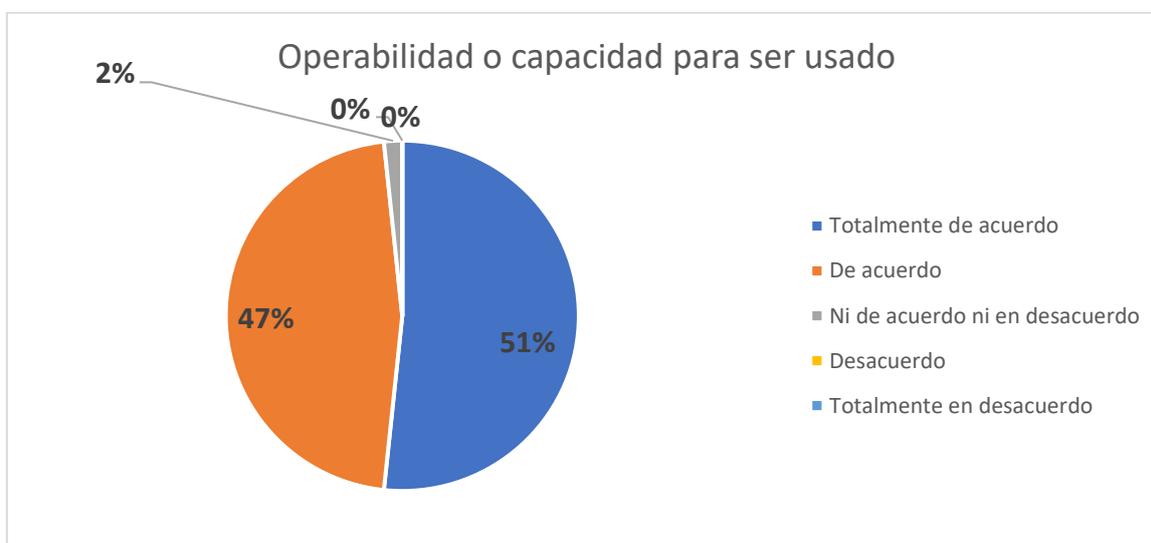


Fig. 59. Gráfico operabilidad o capacidad para ser usado.

Grado de accesibilidad.

En cuanto al grado de accesibilidad, el 57% de los encuestados afirman estar totalmente de acuerdo, y el 43% opinan estar de acuerdo. Así lo muestra el siguiente gráfico:

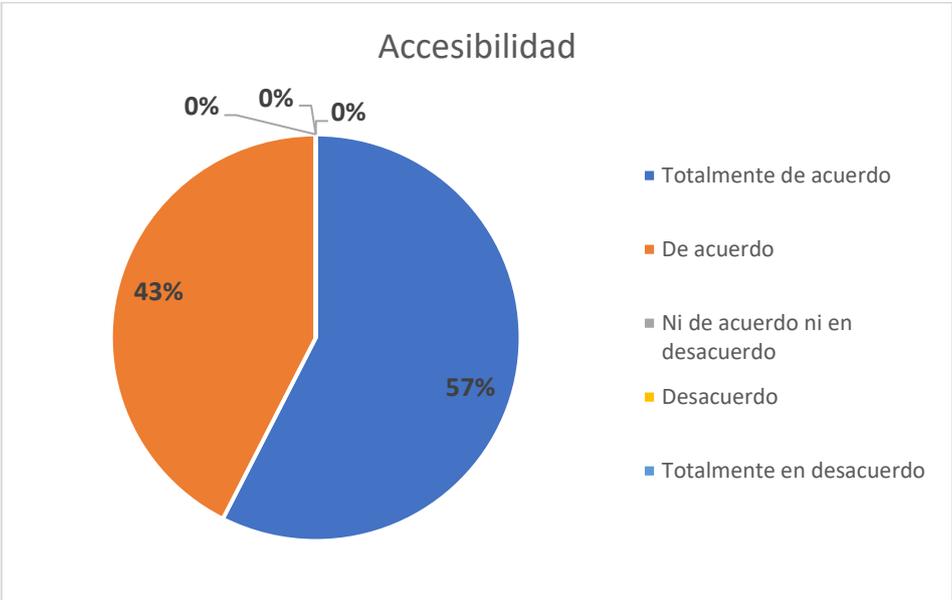


Fig. 60. Gráfico de grado de accesibilidad.

Grado de capacidad de aprendizaje.

Referente al grado de capacidad de aprendizaje, el 63% de los encuestados coincidieron en que están totalmente de acuerdo, el 31% mencionan estar de acuerdo y sólo el 6% de los mismos apuntan estar ni de acuerdo ni en desacuerdo. Así lo muestra el siguiente gráfico:

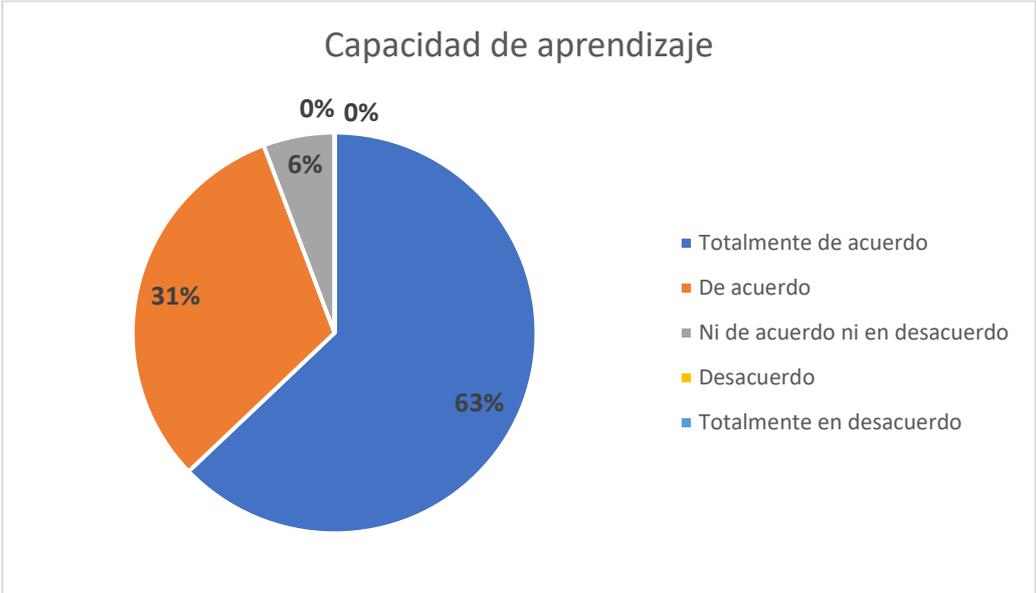


Fig. 61. Gráfico de capacidad de aprendizaje.

3.2.3.2 TIEMPO

Luego del procesamiento de los datos de la dimensión tiempo, se presenta los resultados a continuación:

Tiempo empleado para registrar un incidente.

Referente al tiempo empleado para registrar un incidente, se tiene que el tiempo empleado antes del uso del sistema es 1,902 min en promedio y representa el 100%, así mismo, el tiempo empleado después del uso del sistema es 0,728 min en promedio y representa el 38,28%. Por lo tanto, se observa que existe una diferencia de 1,174 min, lo que se traduce como reducción del 61,72%, tal como se muestra en el siguiente grafico:

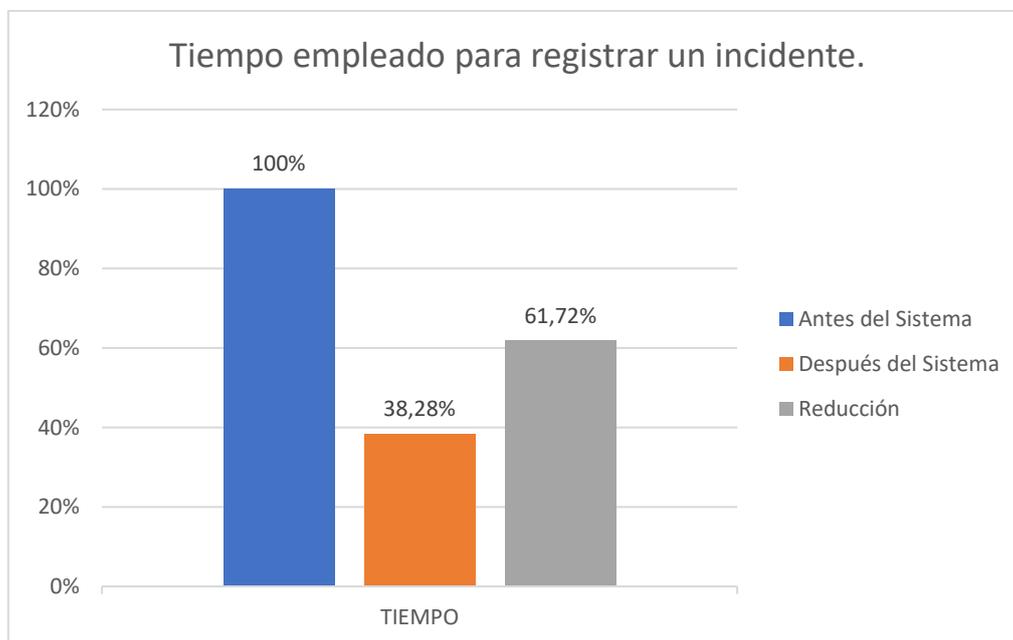


Fig. 62. Gráfico de tiempo empleado para registrar un incidente.

Tiempo empleado para asignar un incidente.

Con respecto al tiempo empleado para asignar un incidente, se tiene que el tiempo empleado antes del uso del sistema es 1,254 min en promedio y representa el 100%, así mismo, el tiempo empleado después del uso del sistema es 0,138 min en promedio y representa el 11,06%. Por lo tanto, se observa que existe una diferencia de 1,11 min, lo que se traduce como una reducción de 88,94%, tal como se muestra en el siguiente grafico:

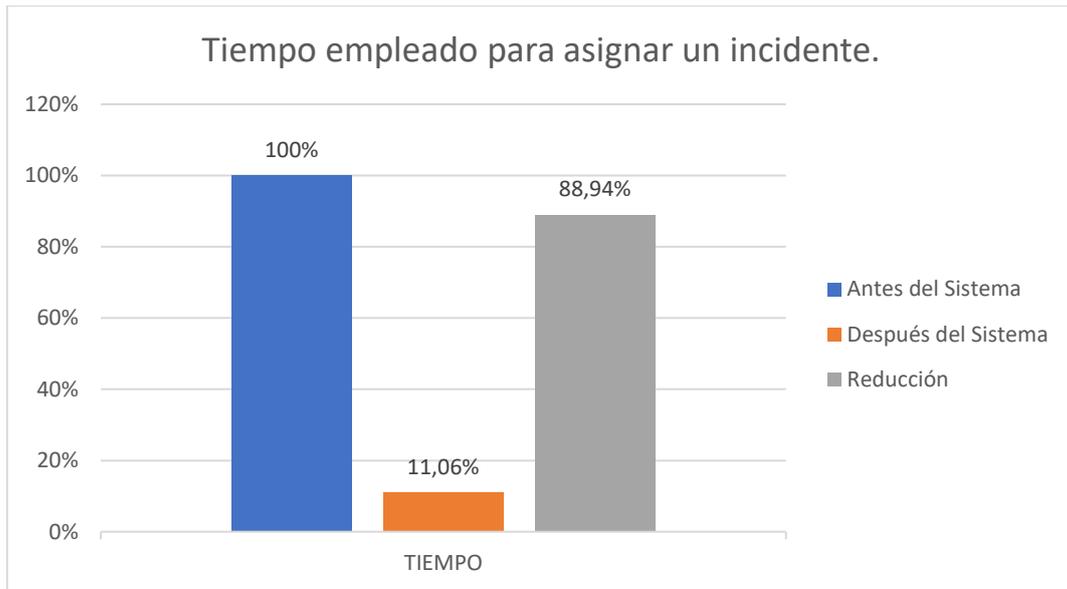


Fig. 63. Gráfico de tiempo empleado para asignar un incidente.

Tiempo empleado para la elaboración de reportes.

Respecto al tiempo empleado para la elaboración de reportes, se tiene que el tiempo empleado antes del uso del sistema es 9,312 min en promedio y representa el 100%, así mismo, el tiempo empleado después del uso del sistema es 0,26 min en promedio y representa el 2,79%. Por lo tanto, se observa que existe una diferencia de 9,052 min, lo que se traduce como reducción del 97,21%, tal como se muestra en el siguiente gráfico:

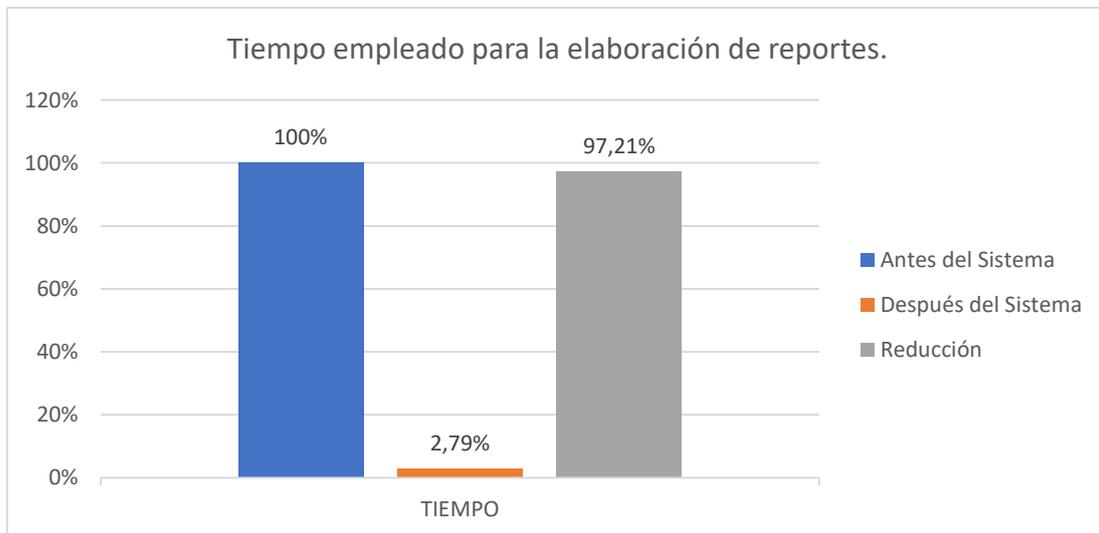


Fig. 64. Gráfico de tiempo empleado para la elaboración de reportes.

3.2.3.3 SATISFACCIÓN DEL EQUIPO DE SOPORTE

Nivel de satisfacción con respecto al proceso de gestión de incidentes.

En relación al nivel de satisfacción con respecto al proceso de gestión de incidentes, se tiene que el nivel de satisfacción antes del uso del sistema es 2,18 en promedio, así mismo, el nivel de satisfacción después del uso del sistema es 4,0 en promedio. Por lo tanto, se observa que existe una variación positiva de 83% en el nivel de satisfacción, tal como se muestra en el siguiente grafico:

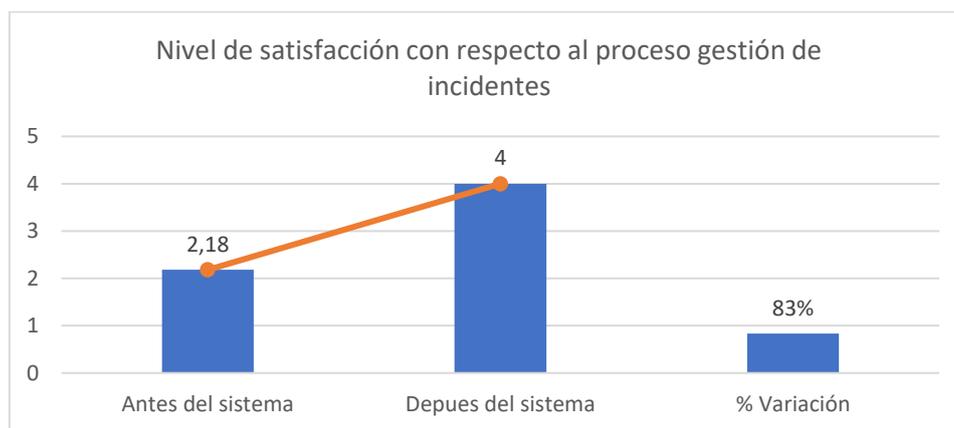


Fig. 65. Gráfico de nivel de satisfacción respecto al proceso.

Nivel de satisfacción con respecto a los tiempos del proceso de gestión de incidentes.

En cuanto al nivel de satisfacción con respecto a los tiempos del proceso de gestión de incidentes, se tiene que el nivel de satisfacción antes del uso del sistema es 2,32 en promedio, así mismo, el nivel de satisfacción después del uso del sistema es 4,40 en promedio. Por lo tanto, se observa que existe una variación positiva de 90% en el nivel de satisfacción, tal como se muestra en el siguiente grafico:

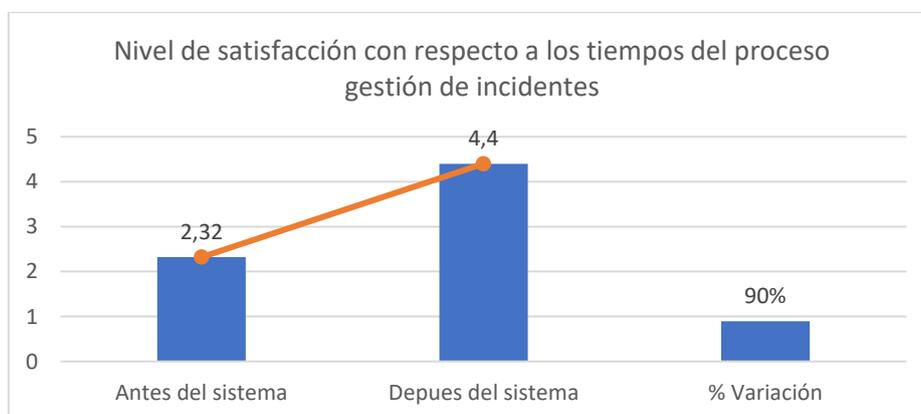


Fig. 66. Gráfico de nivel de satisfacción respecto a los tiempos del proceso.

3.2.3.4 PRODUCTIVIDAD

Número de incidentes atendido por persona del equipo de soporte durante un día de trabajo.

Con respecto a la productividad, se basó en el cálculo del número de incidentes atendidos por personal de equipo de soporte, donde se tiene que el promedio de atención es de 2,68 tickets o incidentes atendidos antes de la implementación de sistema, y 3,32 en promedio después de la implementación del sistema. Por lo tanto, se observa que existe una variación positiva de 24% en el nivel de productividad, tal como se muestra en el siguiente grafico:

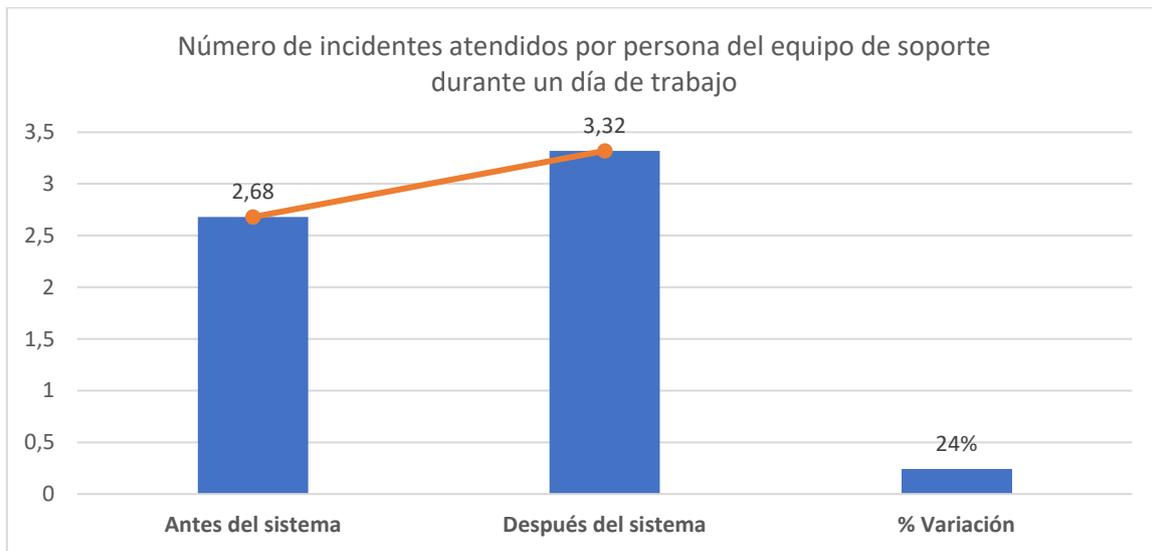


Fig. 67. Gráfico de productividad del equipo de soporte.

CAPÍTULO IV. ANÁLISIS Y DISCUSIÓN DE RESULTADOS

A partir de los hallazgos obtenidos, se acepta la hipótesis general que establece que la implementación de un Sistema Help Desk mejora la gestión de incidentes informáticos en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca. Así mismo, se estudió el problema desde diferentes puntos tales como tiempo, productividad y satisfacción del equipo de soporte.

Respecto al tiempo, se analizaron tres indicadores, donde los resultados generados demuestran que la implementación de un Sistema Help Desk disminuye el tiempo empleado para registrar y asignar un incidente y para elaborar reportes, tal como se muestra en la siguiente tabla:

Tabla XXXVI. Dimensión tiempo - discusión de resultados.

Indicador	Antes (min)	Después (min)	Variación	
Tiempo empleado para registrar de un incidente.	1,902	0,728	1.174	61,72%
Tiempo empleado para asignar un incidente.	1,254	0,138	1,116	88,94%
Tiempo empleado para la elaboración reportes.	9,312	0,26	9,052	97,21%

Estos resultados coinciden con lo que sostienen los autores Ponce y Samaniego [8] en su investigación sobre el impacto del help desk en los procesos de soporte técnico en una organización, donde obtiene como resultado general una mejora del 40% en sus tiempos. Por su parte, Hoyos [10], en su tesis que abarca la implementación de un sistema help desk web en una municipalidad, resalta que existe una reducción del 80,65% en el tiempo de registro de un incidente. Del mismo modo, Cruz [11], en su trabajo de investigación se enfoca en optimizar el proceso de control de evento de tipo informático en una institución mediante la implementación de un sistema help desk, donde obtiene como resultado una reducción del 56% en el tiempo de registro de un incidente. Lo cual es acorde al resultado en la presente investigación.

En cuanto a la satisfacción del equipo de soporte, los dos indicadores analizados demuestran que la implementación de un sistema Help Desk mejora la satisfacción de los usuarios del equipo, tal como se muestra en la siguiente tabla:

Tabla XXXVII. Dimensión satisfacción - discusión de resultados.

Indicador	Antes (1-5)	Después (1-5)	Variación	
Nivel de satisfacción con respecto al proceso de gestión de incidentes.	2,18	4,0	1,82	83%
Nivel de satisfacción con respecto a los tiempos del proceso gestión de incidentes.	2,32	4,4	2,08	90%

Esto es avalado con lo que sostiene Hoyos [10] en su trabajo de investigación en la que propone la implementación de un sistema informático help desk para mejorar el control de incidencias en una municipalidad, que va desde su reporte hasta su finalización; en donde enfatiza que, tras la implementación del sistema, hay un incremento del 22% en el nivel de satisfacción. Cabe resaltar que, en la investigación de Hoyos, la institución ya contaba con algunos procesos automatizados, por lo que el incremento en el nivel de satisfacción es un poco menor en relación al presente trabajo de investigación. Por su parte, Ibáñez [13], también analiza la satisfacción en su tesis, donde propone implementar gestión de incidentes utilizando el marco ITIL, y como resultado muestra una mejora del 70% en el nivel de satisfacción de los usuarios. También, Chavarry y Gallardo [18], mencionan que luego de implementar un sistema help desk, la satisfacción de los trabajadores se ha incrementado.

En lo que corresponde a productividad, se analizó un indicador, donde los resultados muestran que la implementación de un sistema Help Desk mejora la productividad en la Unidad de informática y Sistemas, tal como se puede apreciar en la siguiente tabla:

Tabla XXXVIII. Dimensión productividad - discusión de resultados.

Indicador	Antes (#incidentes)	Después (#incidentes)	Variación	
Número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo.	2,68	3,32	0,64	24%

Estos datos guardan relación con los resultados que obtuvo López [9], en su proyecto que implementa un sistema help desk en una organización, donde menciona que luego de aplicar la ejecución del sistema, existe un incremento del 99% en su capacidad de atención es decir el equipo ha mejorado su productividad. De manera similar, Cruz [11], trata a la productividad basándose en el nivel de acumulación de incidencias, donde una vez aplicado su propuesta de implementar un sistema help desk para mejorar el control de incidencias, obtiene una reducción

en un 50% de esta acumulación lo que permite ver que la productividad ha mejorado. Por su parte, Ibáñez [19], argumenta que la gestión de incidentes de TI ha mejorado considerablemente, ya que luego del uso de un sistema de apoyo a dicho proceso, su capacidad de atención de incidentes se ha incrementado en 113%.

Finalmente, en atención a la variable independiente, se analizó el indicador para medir la usabilidad del sistema help desk teniendo en cuenta el estándar de calidad ISO 25000, en ese sentido se obtuvo los siguientes resultados:

Tabla XXXIX. Dimensión usabilidad - discusión de resultados.

Indicador	Totalmente de acuerdo	De acuerdo	Ni de acuerdo ni desacuerdo
Grado de operabilidad	51%	47%	2%
Grado de accesibilidad	57%	43%	0%
Grado de capacidad de aprendizaje	63%	31%	6%

Dado los resultados mostrados anteriormente, se puede apreciar que el sistema help desk implementado en el presente trabajo de investigación cumple con los requisitos propuestos por el estándar de calidad ISO 25000, lo que significa que el software es fácil de usar, es fácil de operar y controlar y es accesible de cara al usuario.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Tras el análisis efectuado, se concluye que la implementación de un Sistema Help Desk en la gestión de incidentes informáticos en la Unidad de Informática y Sistemas de la Municipalidad Provincial de Cajamarca tiene un impacto positivo, por cuanto presenta una notable mejoría en los puntos analizados, tales como, una reducción promedio del 82,62% en el tiempo para realizar actividades en el proceso de gestión de incidentes, un aumento promedio del 86,50% en el nivel de satisfacción del equipo o trabajadores del área de soporte y un aumento del 24% en el nivel de productividad.

Se identificó el impacto de la implementación de un Sistema Help Desk en el tiempo para realizar las actividades del proceso, resultando una disminución del 61,72% en el tiempo empleado para registrar un incidente, una disminución del 88,94% en el tiempo para asignar un incidente y una disminución del 97,21% en el tiempo empleado para elaborar reportes en la Unidad de informática y Sistemas de la MPC.

También, se determinó el impacto de la implementación de un Sistema Help Desk en la satisfacción del equipo de soporte, donde resulta que existe un aumento del 83% en nivel de satisfacción con respecto al proceso de gestión de incidentes y 90% con respecto a los tiempos del proceso gestión de incidentes al de la Unidad de informática y Sistemas de la MPC.

Además, se dio a conocer el impacto de la implementación de un Sistema Help Desk en la productividad, dejando como resultado una variación positiva del 24% en el número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo.

El Sistema Help Desk ha demostrado ser de gran utilidad para la toma de decisiones, dado que con la data recolectada ha permitido establecer medidas para mantener la infraestructura tecnológica en buen funcionamiento, estableciendo mantenimientos preventivos de los equipos de cómputo, cambio de equipos de conectividad (switch, router) debido a constantes reportes de fallas en la conexión, entre otros.

Respecto a la metodología empleada, se concluye que Scrum y el desarrollo incremental se ajusta al trabajo realizado de desarrollo de un software debido a que es una estrategia de

desarrollo iterativo, donde el producto está en continua evolución y mejora, adaptándose a los cambios de manera flexible, permitiendo implementarlos a lo largo del desarrollo del sistema.

Por último, las herramientas utilizadas para el desarrollo del software resultaron de gran utilidad; el framework laravel reflejó ser potente y sencillo proporcionando un rendimiento excepcional para aplicaciones web, ya que se enfoca en desarrollar aplicaciones seguras. Además, está desarrollado en el lenguaje PHP, que es de software libre y cuenta con una amplia comunidad de apoyo.

5.2 RECOMENDACIONES

Antes de finalizar, se plantea algunas recomendaciones en base a los resultados y las conclusiones a las que se llegó en la presente investigación.

Con la finalidad de ir actualizando el Sistema Help Desk, se recomienda implementar o agregar la funcionalidad de notificaciones en tiempo real, la integración de esta funcionalidad daría un apoyo significado al proceso de atención de incidentes, dado que el personal de soporte técnico no tendría que revisar constantemente el sistema para revisar si tiene algún ticket o incidentes asignado, sino que el propio sistema le notificaría con una alerta cuando el encargado de la mesa de ayuda registre un incidente y asigne el ticket.

Así mismo, dada la relevancia del tema tratado, se cree conveniente recomendar la realización de una investigación para analizar la influencia o la relación del servicio que ofrece el área de Informática y Sistemas en los trabajadores de todas las áreas de Municipalidad Provincial de Cajamarca.

Finalmente, se recomienda continuar utilizando con mayor énfasis la información generada por el Sistema Help Desk, así como la data recolectada, para establecer directivas de control y mejoras, con la finalidad de mantener el buen funcionamiento de la infraestructura tecnológica de la Municipalidad Provincial de Cajamarca.

REFERENCIAS BIBLIOGRÁFICAS

- [1] E. A. V. Briceño, «Gestiopolis,» 2019. [En línea]. Available: <https://www.gestiopolis.com/sistemas-informacion-importancia-empresa/>. [Último acceso: 5 Julio 2019].
- [2] InGenio, «InGenio Learning,» [En línea]. Available: <https://ingenio.edu.pe/gestion-de-incidentes-itol/>. [Último acceso: Junio 2020].
- [3] A. Soria, «muypymes,» 2016. [En línea]. Available: <https://www.beservices.es/behelp-mantenimiento-informatico-p-32-es>. [Último acceso: 23 Junio 2020].
- [4] DoctorClic, DoctorClic, Abril 2018. [En línea]. Available: <https://www.doctorclic.es/problemas-informaticos-empresa/>. [Último acceso: Junio 2020].
- [5] M. Jimenez y D. Larrouy, «ELPAISECONOMIA,» 10 Julio 2015. [En línea]. Available: https://cincodias.elpais.com/cincodias/2015/07/09/tecnologia/1436467530_190970.html. [Último acceso: 20 Octubre 2020].
- [6] Y. Ordaz, «Milenio 2020,» 16 Junio 2016. [En línea]. Available: <https://www.milenio.com/negocios/fallas-sistema-provocan-perdidas-millonarias-empresas>. [Último acceso: 10 Noviembre 2020].
- [7] F. Gil Mena, «Gestión,» 07 Febrero 2019. [En línea]. Available: <https://gestion.pe/tecnologia/colaboradores-pierden-46-laboral-fallas-computadora-258004-noticia/>. [Último acceso: 20 Octubre 2020].
- [8] J. L. Ponce Huanca y M. F. Samaniego Castro, «Análisis del impacto del help desk en los procesos del departamenteo de soporte técnico en una organización,» Guataquil, 2015.
- [9] F. F. López Vera, «Implementación de un sistema de mesa de ayuda (help desk) para el control de incidencias que se presentan en el gobierno autónomo descentralizado de la provincia de Esmeraldas,» Ecuador, 2014.
- [10] L. L. Hoyos Mendoza, «Sistema Informático Help Desk vía Web y Móvil para mejorar el control de incidencias en la Unidad de Tecnologías de Información de la Municipalidad Distrital de Pacasmayo,» 2015.
- [11] A. H. Cruz Velarde, «Sistema help desk para mejorar el control de incidencias de hardware y software bajo la modalidad open source en la Gerencia Regional de Salud La Libertad,» La Libertad, 2017.
- [12] A. M. Chavarry Castillo y J. Gallardo Chicoma, «Influencia de un sistema de Help Desk en la gestión de incidencias de tecnologías de información, de la Municipalidad Distrital de Llacanora periodo - 2017,» Cajamarca, 2018.
- [13] J. L. Ibañez Herrera, «Impacto de la implementación de gestión de incidentes de TI del Framework ITIL V3 en la Sub-Área de End User Computer en Goldfields La Cima S.A. – Operación minera Cerro Corona,» 2014.
- [14] J. Van Bon, Fundamentos de ITIL V3, Van Haren ed., 2008.
- [15] A. Gonzales, 2016. [En línea]. Available: <http://dspace.unitru.edu.pe/bitstream/handle/UNITRU/9580/CORREA%20GONZALES%2C%20Adolfo%20Alberto.pdf?sequence=1&isAllowed=y>. [Último acceso: Mayo 2020].

- [16] ServiceTonic, «ITIL V3. Incident Management,» 2019. [En línea]. Available: <https://www.servicetonic.com/itil/itil-v3-incident-management/>. [Último acceso: 29 Junio 2020].
- [17] CEUPE, «Centro Europeo de Postgrado,» 2020. [En línea]. Available: <https://www.ceupe.com/blog/flujo-de-la-gestion-de-incidencias.html>. [Último acceso: 24 Junio 2020].
- [18] «ManageEngine ServiceDesk Plus,» [En línea]. Available: <https://www.manageengine.com/latam/service-desk/itil-incident-management/kpis-metricas-gestion-de-incidentes-itil.html>.
- [19] L. Carbajal y E. Vásquez, «Rediseño de procesos de helpdesk para generar valor en empresas proveedoras de servicios de TIC,» Trujillo, 2016.
- [20] J. A. Chulle Chapilliquen, «Diseño de un help desk para mejorar el proceso de soporte en el centro de informática y telecomunicaciones - Universidad Nacional de Piura,» 2018.
- [21] INGENIO LEARNING, «InGenio,» Marzo 2020. [En línea]. Available: <https://ingenio.edu.pe/help-desk-teletrabajo-para-profesionales-de-redes/>. [Último acceso: Junio 2020].
- [22] «PedaNet,» [En línea]. Available: <https://peda.net/kenya/ass/subjects2/computer-studies/form-1/the-computer-system#:~:text=A%20computer%20system%20is%20a,Storage%2C%20Output%20and%20Communication%20devices..> [Último acceso: 2021].
- [23] D. Da Silva, «Blog de Zendesk,» Junio 2020. [En línea]. Available: <https://www.zendesk.com.mx/blog/sistema-help-desk/>. [Último acceso: Julio 2021].
- [24] «CERTUS,» [En línea]. Available: <https://www.certus.edu.pe/blog/consiste-desarrollo-software/>. [Último acceso: Mayo 2021].
- [25] E. G. Maida y J. Pacienza, «Metodologías de desarrollo de software,» Buenos Aires, 2015.
- [26] M. Tena, «Transformación Digital,» BBVA, [En línea]. Available: <https://www.bbva.com/es/metodologia-agile-la-revolucion-las-formas-trabajo/>. [Último acceso: Julio 2021].
- [27] Twproject, «Agile methodology,» Twproject, [En línea]. Available: <https://twproject.com/blog/agile-methodology-advantages-disadvantages-innovative-method/>. [Último acceso: Julio 2021].
- [28] Scrum, «The home of Scrum,» Marzo 2020. [En línea]. Available: <https://www.scrum.org/>. [Último acceso: 2 Marzo 2021].
- [29] E. Abellan, «Wearemarketing,» Marzo 2020. [En línea]. Available: <https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>. [Último acceso: Julio 2021].
- [30] ISO25000, «ISO25000 Calidad de software y datos,» [En línea]. Available: <https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>. [Último acceso: 2021].
- [31] B. A. Naranjo Sánchez, M. J. Tinoco Arichavala y D. E. Vega Bravo, «Análisis de la usabilidad del sistema web de terapias cognitivas sanamentics,» Guayaquil, 2019.
- [32] «Documentacion TI,» [En línea]. Available: <https://sites.google.com/site/docutecni/tecnologias/uml/diagramas-uml>. [Último acceso: 15 Marzo 2021].

- [33] «Lucidchart.» [En línea]. Available: <https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>. [Último acceso: 15 Marzo 2021].
- [34] D. Stokdyk, «Southern New Hampshire University,» Octubre 2019. [En línea]. Available: <https://www.snhu.edu/about-us/newsroom/stem/what-is-information-technology>. [Último acceso: Diciembre 2020].
- [35] JMP Statistical Discovery LLC, «JMP Statistical Discovery,» [En línea]. Available: [https://www.jmp.com/es_pe/statistics-knowledge-portal/t-test.html#:~:text=Una%20prueba%20t%20\(tambi%C3%A9n%20conocida,grupos%20mediante%20pruebas%20de%20hip%C3%B3tesis..](https://www.jmp.com/es_pe/statistics-knowledge-portal/t-test.html#:~:text=Una%20prueba%20t%20(tambi%C3%A9n%20conocida,grupos%20mediante%20pruebas%20de%20hip%C3%B3tesis..) [Último acceso: Marzo 2022].
- [36] H. Celina Oviedo y A. Campo Arias, «Metología de investigación y lectura de estudios,» *Revista Colombiana de Psiquiatría*, n° 4, p. 577, 2005.
- [37] B. Niebel, Ingeniería industrial, p. 340.

ANEXOS

ANEXO 1: INSTRUMENTOS PARA LA RECOLECCIÓN DE DATOS

1. Encuesta para medir la usabilidad del sistema:

Objetivo: La presente encuesta tiene por objetivo medir la dimensión Usabilidad, con relación al Sistema Help Desk desarrollado para el proceso de gestión de incidentes en la institución.

Consentimiento informado: La participación en este estudio es estrictamente voluntaria. La información que se recoja será confidencial y anónima; la cual no se usará para ningún otro propósito fuera de los de esta investigación.

Instrucciones:

Responda las siguientes preguntas de acuerdo a su percepción, marcando con una (X) dentro del recuadro que crea conveniente. Considerando la escala de 1 a 5, donde:

Totalmente en desacuerdo.	En desacuerdo.	Ni de acuerdo, ni en desacuerdo.	De acuerdo.	Totalmente de acuerdo.
1	2	3	4	5

Tabla XL. Encuesta para medir la usabilidad del sistema.

Pregunta	1	2	3	4	5
Grado de operabilidad o capacidad para ser usado					
1. Los iconos en el sistema son entendibles y fáciles de ubicar.					
2. La información que ofrece el sistema no es explícita y dificulta su entendimiento.					
3. La ubicación de la información es visible y de fácil identificación.					
4. Los usuarios tardarían en aprender a usar el sistema.					
5. Hay un exceso en la cantidad de acciones para lograr un objetivo dentro del sistema.					
6. Se diferencian fácilmente los mensajes de error y de éxito.					
Grado de accesibilidad					
7. Al utilizar diferentes navegadores, por ejemplo, Opera, Firefox, Chrome u otros, la apariencia y funcionalidad del sistema es similar en todos ellos.					
8. Al utilizar diferentes dispositivos, por ejemplo, Tablet, Laptop o Smart Phone, el sistema se adapta adecuadamente.					
9. Considera que para un adecuado manejo del sistema es indispensable contar con conocimiento técnico.					
10. El tamaño y color de la letra permite leer con facilidad.					
Capacidad de aprendizaje					
11. Los nombres de las opciones dentro del sistema son significativos que permiten predecir su acción.					

Pregunta	1	2	3	4	5
12. Me gustaría usar el sistema con frecuencia.					
13. Encontré el sistema complejo.					
14. Con el uso del sistema reduzco mi nivel de productividad.					
15. Los enlaces u opciones dentro del sistema están bien definidos y claros.					
16. Existen elementos en el sistema que dificulten la navegación.					
17. En general considero que el sistema es fácil de usar.					

2. Encuesta para medir la satisfacción del equipo de soporte.

Objetivo: La presente encuesta tiene por objetivo medir la dimensión satisfacción del equipo de soporte con relación al proceso de gestión de incidentes en la institución.

Consentimiento informado: La participación en este estudio es estrictamente voluntaria. La información que se recoja será confidencial y anónima; la cual no se usará para ningún otro propósito fuera de los de esta investigación.

Instrucciones:

Responda las siguientes preguntas de acuerdo a su percepción, marcando con una (X) dentro del recuadro que crea conveniente. Considerando la escala de 1 a 5, donde:

Totalmente insatisfecho.	Insatisfecho.	Ni satisfecho, ni insatisfecho.	Satisfecho.	Totalmente satisfecho.
1	2	3	4	5

Tabla XLI. Encuesta para medir la satisfacción de los usuarios del sistema.

Pregunta	1	2	3	4	5
Satisfacción con respecto al proceso de gestión de incidentes.					
1. ¿Cuán satisfecho se encuentra con la actividad registro de incidente?					
2. ¿Cuán satisfecho se encuentra con las actividades de asignación de incidentes o tickets?					
3. ¿Cuán satisfecho se encuentra con la actividad comunicar al personal elegido sobre el incidente?					
4. ¿Cuán satisfecho se encuentra con la actividad de toma de nota del incidente (para atender)?					
5. ¿Cuán satisfecho se encuentra con las actividades para la elaboración de reportes?					
Satisfacción con respecto a los tiempos del proceso gestión de incidentes.					
6. ¿Cuán satisfecho se encuentra con el tiempo empleado en la actividad registro de incidente?					
7. ¿Cuán satisfecho se encuentra con el tiempo empleado en las actividades de asignación de incidentes o tickets?					

Pregunta	1	2	3	4	5
8. ¿Cuán satisfecho se encuentra con el tiempo empleado en la actividad comunicar al personal elegido sobre el incidente?					
9. ¿Cuán satisfecho se encuentra con el tiempo empleado en la actividad de toma de nota del incidente (para atender)?					
10. ¿Cuán satisfecho se encuentra con el tiempo empleado en las actividades para la elaboración de reportes?					

3. Ficha de análisis documental.

Objetivo: La presente ficha de análisis documental tiene por objetivo obtener información para la dimensión Productividad, con relación al Sistema Help Desk desarrollado para el proceso de gestión de incidentes en la institución.

Tabla XLII. Ficha de análisis documental.

FICHA DE ANÁLISIS DOCUMENTAL N°1	
Institución: Municipalidad Provincial De Cajamarca	
Área: Unidad de Informática y Sistemas	
Objetivo: Obtener los datos referentes número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo.	
Tipo de documento: Electrónico	
Nombre: Reporte diario de atenciones.	
Revisado por: Responsable de la mesa de ayuda.	Fecha:
Ubicación: Bandeja de enviados del correo institucional	
Descripción: Es el documento que el encargado de la mesa de ayuda, presenta todos los días al finalizar la jornada laboral. La finalidad de la misma es que el jefe del área tenga en conocimiento acerca de las atenciones realizadas. Este reporte es presentado vía correo institucional.	
Datos obtenidos:	
<ul style="list-style-type: none"> - Número total de tickets registrados - Número de tickets atendidos por persona 	

4. *Ficha de observación para medir la dimensión tiempo.*

Objetivo: La presente ficha de observación tiene por objetivo obtener información para la dimensión Tiempo, con relación al Sistema Help Desk desarrollado para el proceso de gestión de incidentes en la institución.

Tabla XLIII. Ficha de observación.

FICHA DE OBSERVACIÓN N°1				PRE-TEST	
Institución: Municipalidad Provincial De Cajamarca					
Área: Unidad De Informática Y Sistemas					
Observado por:			Fecha:		
Objetivo: Obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.					
Actividad	Ciclos (minutos)				
	C1	C2	C...	Cn-1	Cn
Registrar un incidente					
Asignar un incidente					
Elaboración de reportes					

ANEXO 2: COEFICIENTE ESTADÍSTICO DE ALFA DE CRONBACH APLICADO A LOS CUESTIONARIOS

1. Cuestionario aplicado a variable independiente – indicador “usabilidad”.

ID	Item01	Item02	Item03	Item04	Item05	Item06	Item07	Item08	Item09	Item10	Item11	Item12	Item13	Item14	Item15	Item16	Item17
1	8	4	4	2	1	4	4	4	4	5	4	4	4	5	3	4	5
2	4	5	3	2	2	5	4	4	5	4	4	5	5	4	4	5	5
3	6	4	5	1	1	4	4	5	5	4	5	4	4	4	3	5	4
4	7	4	5	2	1	4	5	4	5	5	4	5	4	5	4	3	5
5	2	5	4	2	1	4	5	4	4	4	5	5	5	5	5	4	5
6	9	5	5	1	1	4	5	4	5	4	5	4	5	4	5	5	4
7	5	5	4	2	2	5	5	5	4	5	5	5	5	5	5	4	5
8	1	5	4	2	1	5	5	5	5	4	5	5	5	5	5	3	5
9	3	5	5	2	2	4	5	5	5	5	5	5	5	5	4	5	5
10	10	5	4	2	1	5	5	5	5	5	5	5	4	5	5	5	5

Fig. 68. Datos para validar instrumento encuesta aplicado a la variable independiente.

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,757	17

Fig. 69. Coeficiente de Alfa de Cronbach de encuesta aplicado a variable independiente.

Dado el resultado, el coeficiente de Alfa de Cronbach aplicado a los ítems del instrumento, se calculó a través del software SPSS y su resultado es de 0.757, el que según los autores Celina y Campo [36] tiene una confiabilidad aceptable, dado que el valor se encuentra en el rango 0.70 – 0.90. Por tanto, se concluye que la consistencia interna del instrumento utilizado es aceptable y procede su aplicación.

2. Cuestionario aplicado a variable dependiente – indicador “Satisfacción del equipo de soporte”.

	ID	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
1	1	2	4	2	2	3	2	3	4	2	3
2	2	2	5	4	2	4	2	3	2	3	4
3	3	2	5	4	2	4	4	2	4	4	3
4	4	2	4	2	1	4	3	2	2	2	2
5	5	2	4	1	2	4	2	2	2	2	2
6	6	2	4	2	2	3	2	2	2	2	3
7	7	2	5	2	2	3	1	2	2	2	2
8	8	2	3	2	2	3	2	2	2	3	2
9	9	2	3	2	2	2	2	2	2	3	2
10	10	2	3	2	1	2	1	1	2	2	2

Fig. 70. Datos para validar instrumento encuesta aplicado a la variable dependiente.

Estadísticas de fiabilidad	
Alfa de Cronbach	N de elementos
,843	10

Fig. 71. Coeficiente de Alfa de Cronbach de encuesta aplicado a variable dependiente

Dado el resultado, el coeficiente de Alfa de Cronbach aplicado a los ítems del instrumento, se calculó a través del software SPSS y su resultado es de 0.843, el que según los autores Celina y Campo [36] tiene una confiabilidad aceptable, dado que el valor se encuentra en el rango 0.70 – 0.90. Por tanto, se concluye que la consistencia interna del instrumento utilizado es aceptable y procede su aplicación.

ANEXO 3: VALIDACIÓN DE INSTRUMENTOS DE RECOLECCIÓN DE DATOS

1. Validación ficha de observación para “Tiempo”.

FICHA DE VALIDACIÓN DE INSTRUMENTO

I. DATOS GENERALES

I.1. **Experto:** Ing. Carlos Alfonso Pérez Cerna.
 I.2. **Especialidad:** Ingeniería de Sistemas.
 I.3. **Grado académico:** Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones.
 I.4. **Institución:** Municipalidad Provincial de Cajamarca.
 I.5. **Cargo actual:** Supervisor de Gestión de Tecnologías de Información.
 I.6. **Tipo de instrumento:** Fichas de observación.
 I.7. **Fecha:** 04/11/2021

II. TABLA DE VALORACIÓN POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores	x					
2	Formulado con lenguaje apropiado	x					
3	Adecuado para los sujetos en estudio	x					
4	Facilita la prueba de hipótesis	x					
5	Suficiencia para medir la variable	x					
6	Facilita la interpretación del instrumento	x					
7	Acorde al avance de la ciencia y tecnología		x				
8	Expresado en hechos perceptibles	x					
9	Tiene secuencia lógica	x					
10	Basado en aspectos teóricos		x				
	Total	40	8				

Coeficiente de valoración porcentual: c = 96%

III. OBSERVACIONES Y/O RECOMENDACIONES

Sin comentarios y/o observaciones.



Carlos A. Pérez Cerna
 ING. DE SISTEMAS
 CIP: 135232

Firma de experto.

Fig. 72. Validación de experto – ficha de observación.

2. Validación encuesta para “Satisfacción de equipo de soporte”.

FICHA DE VALIDACIÓN DE INSTRUMENTO

I. DATOS GENERALES

I.1. **Experto:** Ing. Carlos Alfonso Pérez Cerna.
 I.2. **Especialidad:** Ingeniería de Sistemas.
 I.3. **Grado académico:** Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones.
 I.4. **Institución:** Municipalidad Provincial de Cajamarca.
 I.5. **Cargo actual:** Supervisor de Gestión de Tecnologías de Información.
 I.6. **Tipo de instrumento:** Cuestionario para Satisfacción.
 I.7. **Fecha:** 04/11/2021

II. TABLA DE VALORACIÓN POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores	x					
2	Formulado con lenguaje apropiado	x					
3	Adecuado para los sujetos en estudio	x					
4	Facilita la prueba de hipótesis	x					
5	Suficiencia para medir la variable	x					
6	Facilita la interpretación del instrumento	x					
7	Acorde al avance de la ciencia y tecnología		x				
8	Expresado en hechos perceptibles	x					
9	Tiene secuencia lógica	x					
10	Basado en aspectos teóricos	x					
	Total	45	4				

Coeficiente de valoración porcentual: c = 98%

III. OBSERVACIONES Y/O RECOMENDACIONES

Sin comentarios y/o observaciones.



Carlos A. Pérez Cerna
 ING. DE SISTEMAS
 CIP: 135232

Firma de experto.

Fig. 73. Validación de experto – cuestionario satisfacción.

1. Validación ficha de análisis documental para “productividad”.

FICHA DE VALIDACIÓN DE INSTRUMENTO

IV. DATOS GENERALES

IV.1. **Experto:** Ing. Carlos Alfonso Pérez Cerna.
 IV.2. **Especialidad:** Ingeniería de Sistemas.
 IV.3. **Grado académico:** Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones.
 IV.4. **Institución:** Municipalidad Provincial de Cajamarca.
 IV.5. **Cargo actual:** Supervisor de Gestión de Tecnologías de Información.
 IV.6. **Tipo de instrumento:** Ficha de análisis documental.
 IV.7. **Fecha:** 04/11/2021

V. TABLA DE VALORACIÓN POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores	x					
2	Formulado con lenguaje apropiado	x					
3	Adecuado para los sujetos en estudio	x					
4	Facilita la prueba de hipótesis	x					
5	Suficiencia para medir la variable	x					
6	Facilita la interpretación del instrumento	x					
7	Acorde al avance de la ciencia y tecnología		x				
8	Expresado en hechos perceptibles	x					
9	Tiene secuencia lógica	x					
10	Basado en aspectos teóricos		x				
	Total	40	8				

Coeficiente de valoración porcentual: c = 96%

VI. OBSERVACIONES Y/O RECOMENDACIONES

Sin comentarios y/o observaciones.



Carlos A. Pérez Cerna
 ING. DE SISTEMAS
 CIP: 135232
 Firma de experto.

Fig. 74. Validación de experto – ficha de análisis documental.

2. Validación encuesta para “Usabilidad”.

FICHA DE VALIDACIÓN DE INSTRUMENTO

I. DATOS GENERALES

I.1. **Experto:** Ing. Carlos Alfonso Pérez Cerna.
 I.2. **Especialidad:** Ingeniería de Sistemas.
 I.3. **Grado académico:** Máster Universitario en Tecnologías, Sistemas y Redes de Comunicaciones.
 I.4. **Institución:** Municipalidad Provincial de Cajamarca.
 I.5. **Cargo actual:** Supervisor de Gestión de Tecnologías de Información.
 I.6. **Tipo de instrumento:** Cuestionario para Usabilidad.
 I.7. **Fecha:** 04/11/2021

II. TABLA DE VALORACIÓN POR EVIDENCIAS

N°	EVIDENCIAS	VALORACION					
		5	4	3	2	1	0
1	Pertinencia de indicadores	x					
2	Formulado con lenguaje apropiado	x					
3	Adecuado para los sujetos en estudio	x					
4	Facilita la prueba de hipótesis	x					
5	Suficiencia para medir la variable		x				
6	Facilita la interpretación del instrumento	x					
7	Acorde al avance de la ciencia y tecnología	x					
8	Expresado en hechos perceptibles	x					
9	Tiene secuencia lógica	x					
10	Basado en aspectos teóricos	x					
	Total	45	4				

Coeficiente de valoración porcentual: $c = 98\%$

III. OBSERVACIONES Y/O RECOMENDACIONES

Sin comentarios y/o observaciones.



Carlos A. Pérez Cerna
 ING. DE SISTEMAS
 C.P: 135232

Firma de experto.

Fig. 75. Validación de experto – cuestionario usabilidad.

ANEXO 3: PRUEBAS DE NORMALIDAD PARA LA DIMENSIÓN TIEMPO

Utilizando el programa SPSS se realizó las pruebas de normalidad de los datos. Dado que las muestras, son menores que 50 es recomendable utilizar el test de Shapiro-Wilk, el cual plantea la hipótesis nula que una muestra proviene de una distribución normal, con un nivel de significancia de 5% ($\alpha = 0.05$) y una hipótesis alternativa que argumenta que la distribución no es normal:

H₀: La distribución es normal

H_a: La distribución no es normal

Donde, además si:

p-valor $\geq \alpha$, se acepta la H₀.

p-valor $\leq \alpha$, se acepta la H_a.

1. Prueba de normalidad para “tiempo empleado para registrar un incidente”.

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
TAS	,163	20	,168	,930	20	,156
TDS	,222	20	,011	,911	20	,067

a. Corrección de significación de Lilliefors

Fig. 76. Prueba de normalidad de tiempo empleado para registrar un incidente.

Dado los resultados en la figura anterior, se observa que el p-valor tanto de TAS (0.156) y TDS (0.067), son mayores que el nivel de significancia (0.05). Por lo tanto acepta la hipótesis nula y se rechaza la hipótesis alternativa, es decir, la muestra tiene un distribución normal. Y por lo tanto, podemos utilizar una prueba paramétrica.

2. Prueba de normalidad para “tiempo empleado para asignar un incidente”.

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
TAS	,121	25	,200 [*]	,972	25	,707
TDS	,126	25	,200 [*]	,954	25	,304

*. Esto es un límite inferior de la significación verdadera.
a. Corrección de significación de Lilliefors

Fig. 77. Prueba de normalidad de tiempo empleado para asignar un incidente.

Dado los resultados en la figura anterior, se observa que el p-valor tanto de TAS (0.707) y TDS (0.304), son mayores que el nivel de significancia (0.05). Por lo tanto acepta la hipótesis nula y se rechaza la hipótesis alternativa, es decir, la muestra tiene una distribución normal. Y por lo tanto, podemos utilizar una prueba paramétrica.

3. Prueba de normalidad para “tiempo empleado para la elaboración de reportes”.

Pruebas de normalidad						
	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
TAS	,230	5	,200 [*]	,952	5	,754
TDS	,219	5	,200 [*]	,868	5	,257

*. Esto es un límite inferior de la significación verdadera.
a. Corrección de significación de Lilliefors

Fig. 78. Prueba de normalidad de tiempo empleado para la elaboración de reportes.

Dado los resultados en la figura anterior, se observa que el p-valor tanto de TAS (0.754) y TDS (0.257), son mayores que el nivel de significancia (0.05). Por lo tanto acepta la hipótesis nula y se rechaza la hipótesis alternativa, es decir, la muestra tiene una distribución normal. Y por lo tanto, podemos utilizar una prueba paramétrica.

ANEXO 4: TABLA DE GENERAL ELECTRIC.

Esta tabla fue desarrollada por la General Electric Company, y sirve de guía para calcular el número de ciclos a observar [37]:

Tiempo de ciclo (minutos)	Número recomendado de ciclos
0.10	200
0.25	100
0.50	60
0.75	40
1.00	30
2.00	20
2.00-5.00	15
5.00-10.00	10
10.00-20.00	8
20.00-40.00	5
40.00 o más	3

Fig. 79. Tabla General Electric.

ANEXO 5: RECOLECCIÓN DE DATOS DIMENSIÓN TIEMPO.

1. Recolección de datos de “Tiempo empleado para registrar un incidente”.

Tabla XLIV. PRE-TEST - Datos recolectados "tiempo empleado para registrar un incidente".

FICHA DE OBSERVACIÓN N°1																PRE-TEST				
Institución: Municipalidad Provincial De Cajamarca																				
Área: Unidad De Informática Y Sistemas																				
Observado por: David Vásquez Cotrina												Fecha: 08/11/2021 – 12/11/2021								
Objetivo: Obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.																				
Actividad	Ciclos (minutos)																			
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
Registrar un incidente	2.50	0.98	1.65	1.45	2.60	2.17	2.67	2.17	2.50	1.83	1.53	1.65	1.35	1.20	1.28	1.42	2.67	2.52	1.93	1.97

Tabla XLV. POST-TEST - Datos recolectados "tiempo empleado para registrar un incidente".

FICHA DE OBSERVACIÓN N°3															POST-TEST					
Institución: Municipalidad Provincial De Cajamarca																				
Área: Unidad De Informática Y Sistemas																				
Observado por: David Vásquez Cotrina										Fecha: 15/02/2022 -18/02/2022										
Objetivo: Obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.																				
Actividad	Ciclos (minutos)																			
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20
Registrar un incidente	1.77	1.08	1.42	1.25	0.42	0.65	0.07	0.63	0.68	0.63	0.85	0.65	0.62	0.65	0.60	0.67	0.68	0.53	0.43	0.28

2. Recolección de datos de “Tiempo empleado para asignar un incidente”.

Tabla XLVI. PRE-TEST - Datos recolectados "tiempo empleado para asignar un incidente".

FICHA DE OBSERVACIÓN N°2										PRE-TEST															
Institución: Municipalidad Provincial De Cajamarca																									
Área: Unidad De Informática Y Sistemas																									
Observado por: David Vásquez Cotrina										Fecha: 08/11/2021 – 12/11/2021															
Objetivo: Obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.																									
Actividad	Ciclos (minutos)																								
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25
Registrar un incidente	0.62	0.98	0.67	0.77	0.67	0.55	0.82	0.97	1.17	0.97	0.53	1.28	1.43	1.17	0.97	0.68	0.72	1.22	0.92	1.00	1.70	0.98	1.11	1.02	1.05

Tabla XLVII.POST-TEST - Datos recolectados "tiempo empleado para asignar un incidente".

FICHA DE OBSERVACIÓN N°4											POST-TEST														
Institución: Municipalidad Provincial De Cajamarca																									
Área: Unidad De Informática Y Sistemas																									
Observado por: David Vásquez Cotrina												Fecha: 15/02/2022 -18/02/2022													
Objetivo: obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.																									
Actividad	Ciclos (minutos)																								
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25
Registrar un incidente	0.15	0.13	0.17	0.13	0.18	0.17	0.20	0.12	0.10	0.13	0.17	0.15	0.18	0.15	0.12	0.20	0.10	0.08	0.17	0.13	0.14	0.14	0.11	0.08	0.10

3. Recolección de datos de “Tiempo empleado para la elaboración de reportes”.

Tabla XLVIII. PRE-TEST - Datos recolectados "tiempo empleado para la elaboración de reportes".

FICHA DE OBSERVACIÓN N°3				PRE-TEST	
Institución: Municipalidad Provincial De Cajamarca					
Área: Unidad De Informática Y Sistemas					
Observado por: David Vásquez Cotrina				Fecha: 08/11/2021–12/11/2021	
Objetivo: obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.					
Actividad	Ciclos (minutos)				
	C1	C2	C3	C4	C5
Elaboración de reportes	8.65	10.9	9.48	8.45	9.08

Tabla XLIX. POST-TEST - Datos recolectados "tiempo empleado para la elaboración de reportes".

FICHA DE OBSERVACIÓN N°6				POST-TEST	
Institución: Municipalidad Provincial De Cajamarca					
Área: Unidad De Informática Y Sistemas					
Observado por: David Vásquez Cotrina				Fecha: 15/02/2022 -18/02/2022	
Objetivo: Obtener los tiempos que toma realizar las actividades de registro de un incidente, asignación de un incidente y elaboración de reportes.					
Actividad	Ciclos (minutos)				
	C1	C2	C3	C4	C5
Elaboración de reportes	0.27	0.28	0.25	0.30	0.20

ANEXO 6: RECOLECCIÓN DE DATOS DIMENSIÓN SATISFACCION DEL EQUIPO DE SOPORTE.

Tabla L. Datos pre-test de encuesta de satisfacción del equipo de soporte.

Datos Satisfacción del equipo de soporte												
Trabajador	Respecto al proceso de gestión de incidentes						Respecto a los tiempos del proceso gestión de incidentes					
	P1	P2	P3	P4	P5	Sum.	P6	P7	P8	P9	P10	Sum.
1	1	1	2	1	1	6	2	1	1	1	1	6
2	2	1	1	1	1	6	1	1	2	1	1	6
3	2	1	1	1	1	6	2	2	1	1	1	7
4	2	2	1	1	1	7	1	1	1	1	1	5
5	1	2	1	2	1	7	1	2	1	1	1	6
6	1	1	1	1	1	5	2	2	1	1	1	7
7	1	1	2	1	1	6	1	1	1	1	1	5
8	1	2	1	1	1	6	2	1	1	1	1	6
9	2	1	2	1	1	7	2	1	1	1	1	6
10	1	1	2	1	1	6	1	1	2	1	1	6
Promedio	1,4	1,3	1,4	1,1	1	6,2	1,5	1,3	1,2	1	1	6

Tabla LI. Datos post-test de encuesta de satisfacción del equipo de soporte.

Datos Satisfacción del equipo de soporte												
Trabajador	Respecto al proceso de gestión de incidentes						Respecto a los tiempos del proceso gestión de incidentes					
	P1	P2	P3	P4	P5	Sum.	P6	P7	P8	P9	P10	Sum.
1	1	2	2	1	1	7	2	1	1	1	1	6
2	2	1	1	1	1	6	1	1	2	1	1	6
3	2	1	1	1	1	6	2	2	1	1	1	7
4	2	2	1	1	1	7	1	1	1	1	1	5
5	1	2	1	2	1	7	1	2	1	1	1	6
6	1	1	1	1	1	5	2	2	1	1	1	7
7	1	1	2	1	1	6	1	1	1	1	1	5
8	1	2	1	1	1	6	2	1	1	1	1	6
9	2	1	2	1	1	7	2	1	1	1	1	6
10	1	1	2	1	1	6	1	1	2	1	1	6
Promedio	1,4	1,4	1,4	1,1	1	6,3	1,5	1,3	1,2	1	1	6

ANEXO 7: RECOLECCIÓN DE DATOS DIMENSIÓN PRODUCTIVIDAD.

Tabla LII. Recolección de datos para dimensión productividad - pretest.

FICHA DE ANÁLISIS DOCUMENTAL N°1						PRE-TEST				
Institución: Municipalidad Provincial De Cajamarca										
Área: Unidad de Informática y Sistemas										
Objetivo: Obtener los datos referentes al número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo.										
Tipo de documento: Electrónico										
Nombre: Reporte diario de atenciones.										
Revisado por: Responsable de la mesa de ayuda.						Fecha: 16/02/2022				
Ubicación: Bandeja de enviados del correo institucional										
Descripción: Es el documento que el encargado de la mesa de ayuda, presenta todos los días al finalizar la jornada laboral. La finalidad de la misma es que el jefe del área tenga en conocimiento acerca de las atenciones realizadas. Este reporte en presentado vía correo institucional.										
Datos obtenidos: Número de tickets atendidos por persona en promedio durante un día de trabajo.										
Trab.	1	2	3	4	5	6	7	8	9	10
Cant.	3,6	2,6	3,8	3,4	3,6	2,6	2,4	1,2	2,2	1,4

Tabla LIII. Recolección de datos para dimensión productividad – post-test.

FICHA DE ANÁLISIS DOCUMENTAL N°2						POST-TEST				
Institución: Municipalidad Provincial De Cajamarca										
Área: Unidad de Informática y Sistemas										
Objetivo: Obtener los datos referentes al número de incidentes atendidos por persona del equipo de soporte durante un día de trabajo.										
Tipo de documento: Electrónico										
Nombre: Reporte diario de atenciones.										
Revisado por: Responsable de la mesa de ayuda.						Fecha: 20/04/2022				
Ubicación: Bandeja de enviados del correo institucional										
Descripción: Es el documento que el encargado de la mesa de ayuda, presenta todos los días al finalizar la jornada laboral. La finalidad de la misma es que el jefe del área tenga en conocimiento acerca de las atenciones realizadas. Este reporte en presentado vía correo institucional.										
Datos obtenidos: Número de tickets atendidos por persona en promedio durante un día de trabajo.										
Trab.	1	2	3	4	5	6	7	8	9	10
Cant.	4	4	4	4,4	3,2	2,8	4,2	1,2	3,8	1,6

ANEXO 8: RECOLECCIÓN DE DATOS VARIABLE INDEPENDIENTE.

Tabla LIV. Datos de variable independiente.

Usabilidad del Sistema																				
Trabajador	Grado de operabilidad o capacidad para ser usado							Grado de accesibilidad					Grado de capacidad de aprendizaje							
	P 1	P 2	P 3	P 4	P 5	P 6	Su m	P 7	P 8	P 9	P 10	Su ma	P 11	P 12	P 13	P 14	P 15	P 16	P 17	Su m
1	5	4	4	5	5	5	28	5	5	4	5	19	5	5	5	5	3	5	5	33
2	5	4	4	5	4	5	27	4	4	4	5	17	5	5	5	5	5	4	5	34
3	5	5	4	4	4	5	27	5	5	5	5	20	5	5	5	5	4	5	5	34
4	5	3	4	4	5	4	25	4	5	4	4	17	4	5	5	4	4	5	5	32
5	5	4	4	4	5	5	27	5	4	5	5	19	5	5	5	5	4	5	4	33
6	4	5	5	5	4	4	27	5	5	4	5	19	4	4	4	4	3	5	4	28
7	4	5	4	5	4	5	27	4	5	5	4	18	5	4	5	4	3	5	5	31
8	4	4	4	5	4	4	25	4	4	4	5	17	4	4	4	5	3	4	5	29
9	5	5	5	5	4	5	29	4	5	4	4	17	5	4	5	4	5	5	4	32
10	5	4	4	5	5	5	28	5	5	5	5	20	5	4	5	5	5	5	5	34
Prom.	4,7	4,3	4,2	4,7	4,4	4,7	27	4,5	4,7	4,4	4,7	18,3	4,7	4,5	4,8	4,6	3,9	4,8	4,7	32

ANEXO 9: PROCESO DE GESTIÓN DE INCIDENTES CON LA APLICACIÓN DEL SISTEMA HELPDESK.

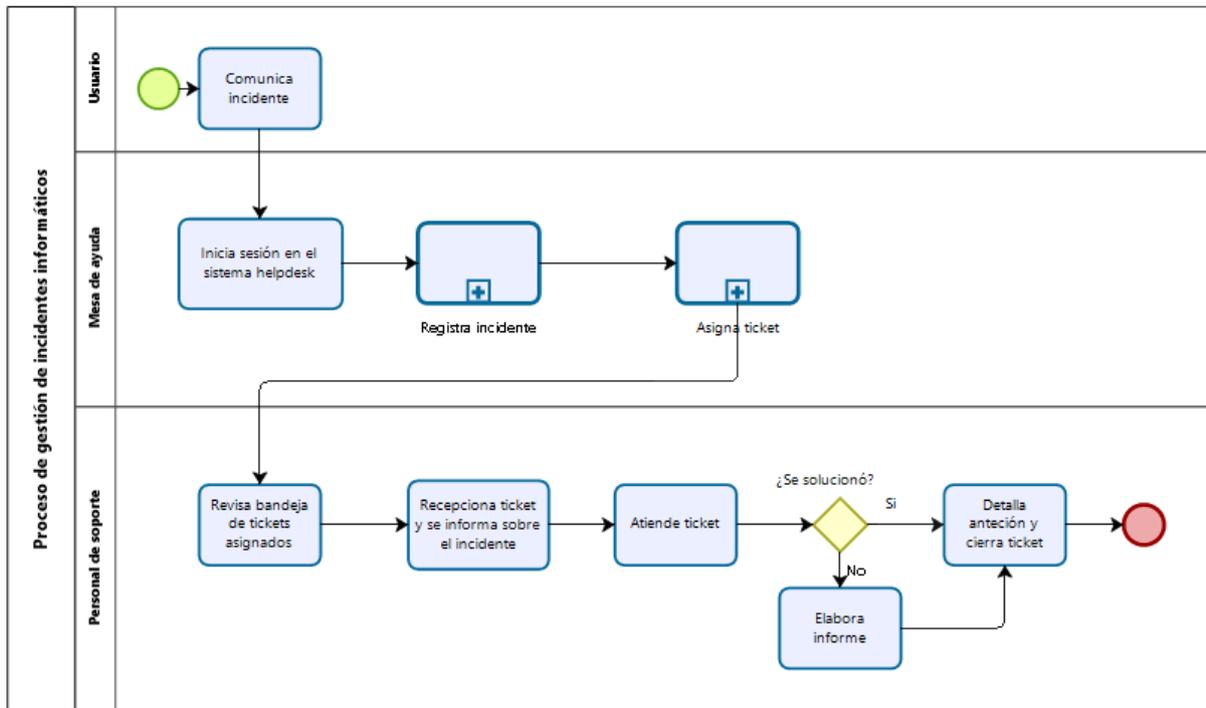


Fig. 80. Nuevo proceso de gestión de incidentes.

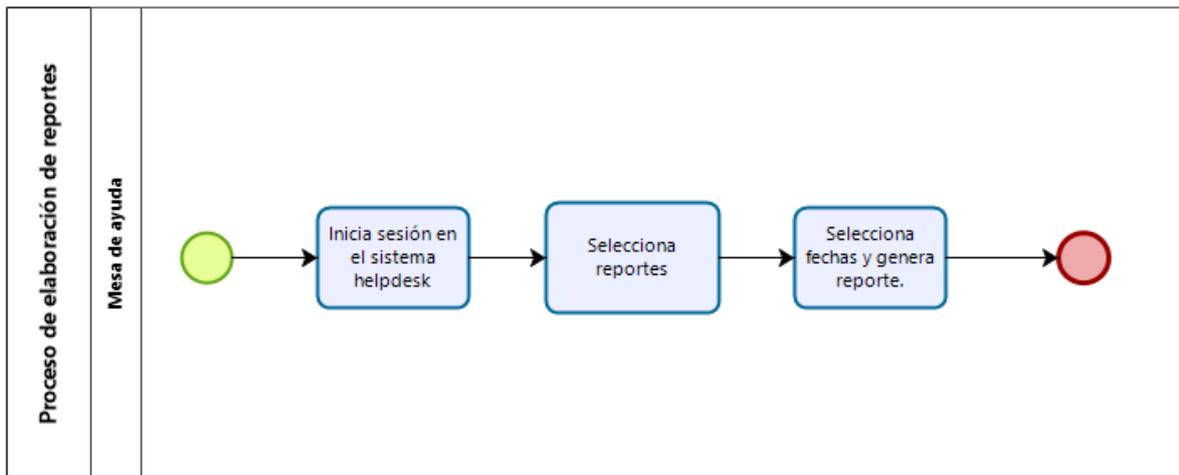


Fig. 81. Nuevo proceso de elaboración de reportes.

ANEXO 10: PRESENTACIÓN DEL SISTEMA HELP DESK.



Fig. 82. Presentación del Sistema Help Desk al cliente.

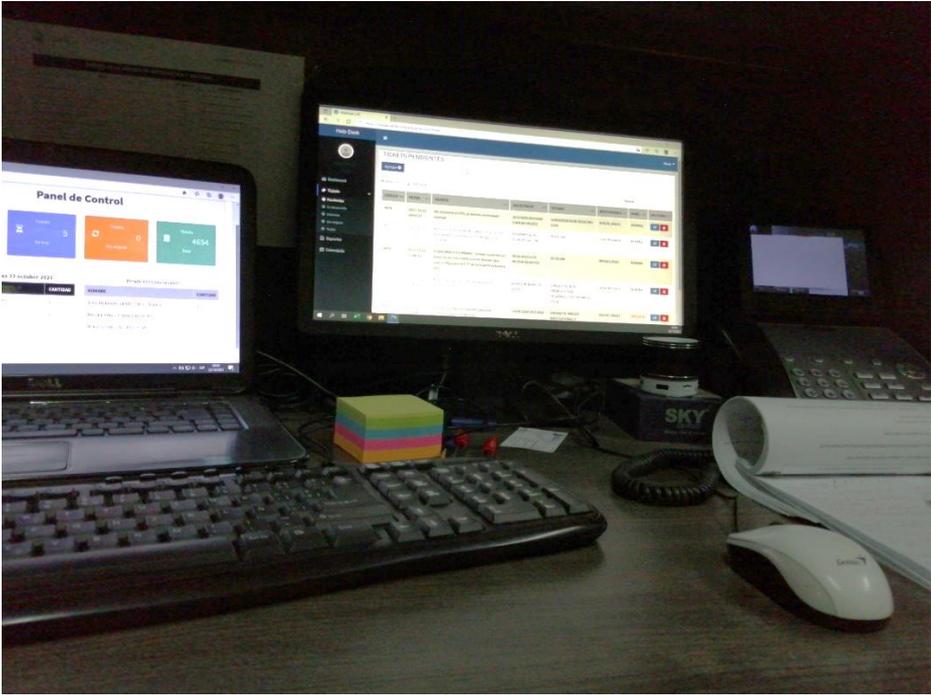


Fig. 83. Sistema Help Desk en funcionamiento.

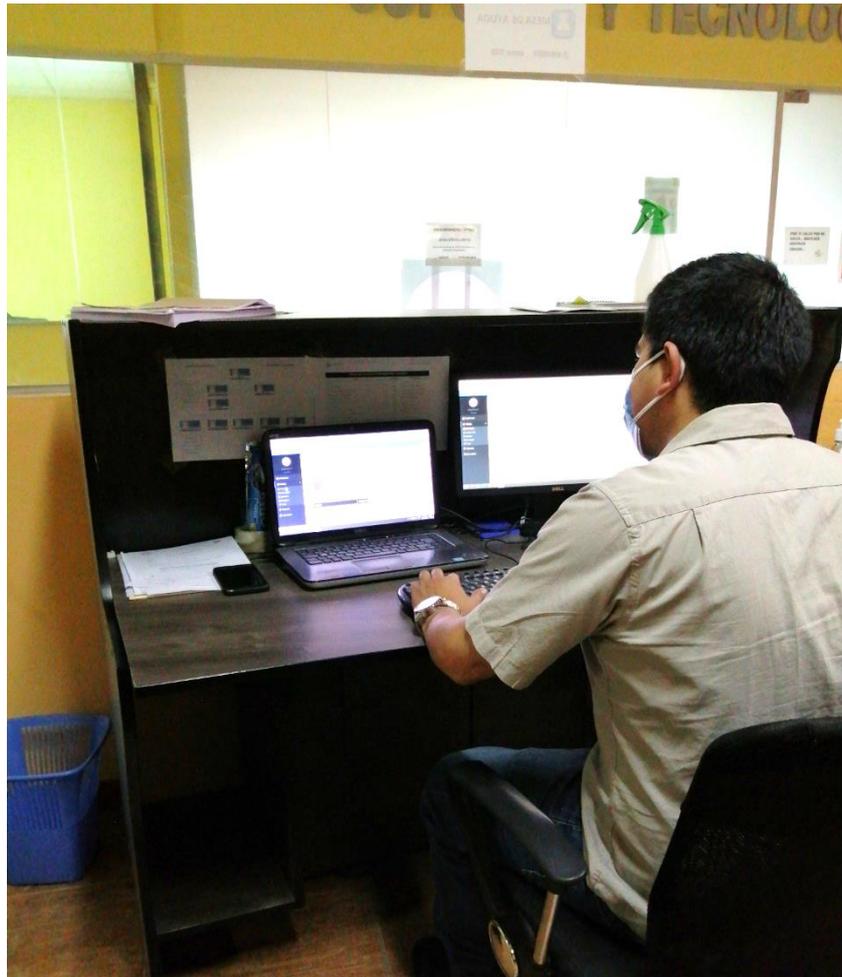


Fig. 84. Encargado de la mesa de ayuda utilizando el sistema help desk.