



DESARROLLO DE UN PROTOTIPO DE APLICACIÓN WEB MÓVIL, PARA DENUNCIAS DE CONTROL URBANO DE OBRAS EN CONSTRUCCIÓN EN LA LOCALIDAD TRES DE LA CIUDAD DE CARTAGENA, APLICANDO BUENAS PRÁCTICAS DE INGENIERÍA DE SOFTWARE (CMMI)

Presentado por:
DEIMER AVILA ARGOTE
BRAYAN BRAVO SILVA

UNIVERSIDAD DEL SINÚ ELÍAS BECHARA ZAINÚM
FACULTAD DE CIENCIAS EXACTAS E INGENIERÍAS
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA-COLOMBIA

2019



DESARROLLO DE UN PROTOTIP
O DE APLICACIÓN WEB MÓVIL, PARA DENUNCIAS DE CONTROL URBANO DE
OBRAS EN CONSTRUCCIÓN EN LA LOCALIDAD 3 DE LA CIUDAD DE CARTAGENA

Trabajo de grado presentado como requisito para optar el título de
INGENIERO DE SISTEMAS

Asesor disciplinar
Luis Alfredo Blanquicett

Asesor metodológico
Eugenia Arrieta Rodríguez

UNIVERSIDAD DEL SINÚ ELÍAS BECHARA ZAINÚM
FACULTAD DE CIENCIAS EXACTAS E INGENIERÍAS
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA-COLOMBIA

2019

Tabla de Contenido

RESUEMEN	10
INTRODUCCION	10
1. DISEÑO METODOLÓGICO	11
1.1 Planteamiento del problema	11
1.1.1 Descripción del problema	11
1.2 Justificación	14
1.4 Objetivos	15
Objetivo general	15
Objetivos específicos	16
1.5 Alcance del proyecto	16
1.6 Estado del arte	19
1.7 Marco referencial	21
Marco teórico	21
Marco conceptual	27
Marco legal	28
Selección del modelo y metodología	29
2. ANÁLISIS DEL SISTEMA	34
2.1 Alcance del proyecto	34
Características de la aplicación	37
Funciones del proyecto	38
Características del usuario.	40
Restricciones generales	40
Suposiciones y dependencias	40
Requisitos del sistema	41
Requisitos Funcionales	42
3. DISEÑO DEL SISTEMA	42
3.1 Diagrama de secuencia	58
Diagrama de secuencia: Ciudadano	5

Diagrama de secuencia: Administrador	59
Diagrama de secuencia: Docente	60
3.2 Diagrama de procesos	61
Diagrama de procesos, modulo administrador	61
Diagrama de procesos, módulo ciudadano	62
Diagrama de procesos, módulo docente	63
3.3 Diagrama de componentes	64
3.4 Diagrama de despliegue	65
3.5 Diagrama de distribución	66
3.6 Diagrama de clases	67
Entidad relación	67
Modelo relacional	68
Diagrama relacional	69
3.7 Diseño de interfaces de usuario	70
Inicio de sesión	70
Registro de usuario	73
Recuperar contraseña	75
Menú principal	77
Crear incidencia	79
Mis denuncias	82
Vista de denuncias	84
Información general	86
Página principal	87
Gestionar denuncia	88
Agregar docente	90
Agregar administrador	92
Generar Reporte	94
Mapa de incidencias	96
Página Principal	96
Lista de casos	98
Notas del caso	100

Mapa de incidencias	102
Cambio de contraseña	104
4. DESARROLLO DEL PROTOTIPO	106
4.1 Tecnologías y lenguaje	106
Modelo de desarrollo	108
Métricas de desarrollo	108
Clases	112
Funciones y métodos	113
4.2 Módulos del sistema	116
Modulo ciudadano	116
Módulo administrador	139
Módulo docente	155
5. PRUEBAS DEL SISTEMA	163
5.1 Pruebas de interfaz	163
Resultado Final	163
5.2 Módulo administrador	166
Resultado final	168
5.3 Módulo docente	169
Resultado final	172
5.4 Pruebas Funcionales	173
Ciclo 1.0, Pruebas funcionales con relación a los módulos, Ciudadanos, Administrador y Docente	173
Módulo ciudadano	173
Módulo administrador	176
Módulo docente	179
Ciclo 2.0, pruebas funcionales con relación a los módulos: Ciudadano, Administrador y docente	183
Modelo ciudadano	183
Módulo administrador	187
Modulo docente	189
Ciclo 3.0, pruebas funcionales con relación a los módulos Ciudadano, Administrador y Docente	193

Módulo ciudadano	193
Módulo Administrador	197
Módulo Docente	200
5.5 Pruebas de severidad	204
Ciclo 1.0	204
Módulo Administrador	204
Módulo ciudadano	207
Módulo docente	211
Ciclo 2.0	215
Módulo Administrador	216
Módulo Ciudadano	222
Módulo Docente	228
Ciclo 3.0	237
Módulo Administrador	238
Módulos Ciudadano	240
Módulo Docente	243
6. CONCLUSIONES Y RECOMENDACIONES	247
7. REFERENCIAS	250
ANEXOS	253
Anexo A: Planeación del proyecto (PP)	253
Anexo B: Gestión de requerimientos (RD)	253
Anexo C: Gestión de requisitos (REQM)	253
Anexo D. Integración del producto (PI)	253
Anexo E: Verificación (VER)	253
Anexo F: Validación (VAL)	253

Lista de tablas

- Tabla 1. Involucrados Relevantes
- Tabla 2. Funciones del proyecto, Fuente Autor
- Tabla 3. Requerimientos. Fuente Autor
- Tabla 4. Registro usuario ciudadano, Fuente Autor
- Tabla 5. Autenticar Usuario, Fuente Autor
- Tabla 6. Modificar Usuario, Fuente Autor
- Tabla 7. Diligenciamiento de denuncia, Fuente autor.
- Tabla 8. Gestión de la denuncia, Fuente Autor
- Tabla 9. Registro de Imagen, Fuente autor
- Tabla 10. Registrar Geolocalización, Fuente autor
- Tabla 11. Consulta de incidencias, Fuente autor
- Tabla 12. Asignar Incidencias, Fuente autor.
- Tabla 13. Lista de Incidencias, Fuente Autor
- Tabla 14. Registro de usuario Administrador, Fuente Autor
- Tabla 15. Modificación en cuestionario de denuncia, Fuente Autor
- Tabla 16. Autenticación de usuario Docente, Fuente Autor
- Tabla 17. Consultar asignaciones de incidencias, Fuente autor
- Tabla 18. Generar Reportar sobre incidencias, Fuente autor.
- Tabla 19. Chat Bot, Fuente Autor
- Tabla 20. Inicio de sesión, formato REQ_VIEW_V1
- Tabla 21. Registro de usuario, formato REQ_VIEW_V1
- Tabla 22. Recuperar Contraseña, formato REQ_VIEW_V1
- Tabla 23. Menú principal, formato REQ_VIEW_V1
- Tabla 24. Nueva incidencia, formato REQ_VIEW_V1
- Tabla 25. Cargar imágenes, formato REQ_VIEW_V1
- Tabla 26. Mis denuncias, formato REQ_VIEW_V1
- Tabla 27. Vista de la denuncia, formato REQ_VIEW_V1
- Tabla 28. Opciones generales, formato REQ_VIEW_V1

- Tabla 29. Asignar caso a Docente, formato REQ_VIEW_V1
- Tabla 30. Asignar caso a Docente, formato REQ_VIEW_V1
- Tabla 31. Agregar administrador, formato REQ_VIEW_V1
- Tabla 32. Generar Reporte, formato REQ_VIEW_V1
- Tabla 33. Mapa de incidencias, formato REQ_VIEW_V1
- Tabla 34. Lista de casos, formato REQ_VIEW_V1
- Tabla 35. Notas del caso, formato REQ_VIEW_V1
- Tabla 36. Mapa de denuncias, formato REQ_VIEW_V1
- Tabla 37. Cambio de contraseña, formato REQ_VIEW_V1
- Tabla 38. Pruebas de interfaz gráfica ciudadano, Fuente autor
- Tabla 39. Informe resultado de prueba funcional V 1.0 Modulo Ciudadano
Elaborado por los autores
- Tabla 40. Informe resultado de prueba funcional V 1.0 Modulo Administrador
Elaborado por los autores
- Tabla 41. Informe resultado de prueba funcional V 2.0 Modulo Docente
Elaborado por los autores
- Tabla 42. Informe resultado de prueba funcional V 2.0 Modulo Ciudadano
Elaborado por los autores, Fuente autor
- Tabla 43. Informe resultado de prueba funcional V 2.0 Modulo Administrador,
fuente autor
- Tabla 44. Informe resultado de prueba funcional V 2.0 Modulo Docente
Elaborado por los autores
- Tabla 45. Informe resultado de prueba funcional V 3.0 Modulo Usuario Elaborado
por los autores
- Tabla 46. Informe resultado de prueba funcional V 3.0 Modulo Administrador
Elaborado por los autores
- Tabla 46. Informe resultado de prueba funcional V 3.0 Modulo Docente
Elaborado por los autores
- Tabla 47. Pruebas de severidad, fuente autor
- Tabla 48. Pruebas de severidad, fuente autor

- Tabla 49. Pruebas de severidad, fuente autor
- Tabla 50. Pruebas de severidad, fuente autor
- Tabla 51, Pruebas de severidad, fuente autor
- Tabla 52, Pruebas de severidad, fuente autor
- Tabla 53. Ciclo 3, prueba de severidad, fuente autor
- Tabla 54. Ciclo 3, prueba de severidad, fuente autor

Tabla de ilustraciones

Ilustración 1. POO, fuente code.google.com	24
Ilustración 2. Código manual PHP, fuente Manual php	25
Ilustración 3. Ejemplo UML, Fuente code.google.com	28
Ilustración 4. Rad Model	32
Ilustración 5. Fases del capability maturity model integration (cmmi), Fuente autor.	35
Ilustración 6. Características de usuario, Fuente autor	42
Ilustración 7. Caso de un uso ciudadano, Fuente autor	58
Ilustración 8, Caso de uso Administrador. Fuente autor	59
Ilustración 9. Caso de uso Docente, Fuente autor	60
Ilustración 10. . Diagrama de secuencia Ciudadano, Fuente autor	61
Ilustración 11. Diagrama de secuencia Administrador, Fuente autor	62
Ilustración 12. Diagrama de secuencia Docente, Fuente autor	63
Ilustración 13. Diagrama de secuencia Administrador, Fuente autor	64
Ilustración 14. Diagrama de secuencia Ciudadano, Fuente autor	65
Ilustración 15 Diagrama de secuencia Docente, Fuente autor	66
Ilustración 16. Diagrama de componentes, Fuente autor	67
Ilustración 17. Diagrama de despliegue, Fuente autor	68
Ilustración 18. Diagrama de distribución, Fuente autor	69
Ilustración 19. Modelo Entidad Relación, Fuente autor	70
Ilustración 20. Modelo Relacional, fuente autor	71
Ilustración 21. Diagrama Relacional, fuente Autor	72
Ilustración 22. Inicio de sesión, fuente autor	73

Ilustración 23. Registro de usuario, fuente autor	75
Ilustración 24. Recuperar Contraseña, fuente autor	78
Ilustración 25. Menú principal, fuente autor	80
Ilustración 26. Nueva incidencia, fuente autor	82
Ilustración 27. Cargar imágenes, fuente autor	83
Ilustración 28. Mis denuncias, fuente autor	85
Ilustración 29. Consulta de incidencias, fuente autor	86
Ilustración 30. Vista de la denuncia, fuente autor	87
Ilustración 31. Opciones generales, fuente autor	88
Ilustración 32. Vista Principal Administrador, fuente autor	90
Ilustración 33. Asignar caso a Docente, fuente autor	91
Ilustración 34. Asignar caso a Docente, fuente autor	93
Ilustración 35. Solicitud de acceso administrador, fuente autor	94
Ilustración 36. Agregar administrador, fuente autor	95
Ilustración 37. Generar Reporte, fuente autor	97
Ilustración 38. Mapa de incidencias, fuente autor	98
Ilustración 39. Página principal, fuente autor	99
Ilustración 40. Lista de casos, fuente autor	100
Ilustración 41. Notas del caso, fuente autor	102
Ilustración 42. Mapa de denuncias, fuente autor	104
Ilustración 43. Cambio de contraseña, Fuente autor	107
Ilustración 46. Diagrama de componentes módulo Ciudadano, Fuente Autor	120
Ilustración 47. Mockup Loging, Fuente autor	121
Ilustración 48. Vista Frontend login, Fuente autor	121
Ilustración 49. Codificación, Fuente Autor	123
Ilustración 50. Mockup Registro de usuario, fuente autor	124
Ilustración 51. vista frontend registro de usuario, fuente autor	125
Ilustración 52. Campos de usuario, Fuente autor	125
Ilustración 53. Mockup vista principal ciudadano, fuente autor	127
Ilustración 54. Frontend vista página principal ciudadano, fuente autor	129
Ilustración 55. Mockup diligenciar denuncia, fuente autor	131
Ilustración 56. Fuente autor	132
Ilustración 57. Agregar Imágenes, Fuente autor	132
Ilustración 58. Vista front nueva denuncia, fuente autor	134
Ilustración 59. Vista front nueva denuncia, fuente autor	136
Ilustración 60. Mockup consultar denuncia, fuente autor	137
Ilustración 61. Vista frontend consultar denuncia, fuente autor	139
Ilustración 62. Vista de Información general, fuente autor	140
Ilustración 63. Vista Frontend de información general, fuente Autor	141
Ilustración 64. Diagrama de componentes Administrador, Fuente Autor	142
Ilustración 65. Vista principal administrador, fuente autor:	143

Ilustración 66. Vista frontend página principal-Administrador. Fuente Autor	145
Ilustración 67. Mockup asignación de denuncias, fuente autor	146
Ilustración 68. Asignar denuncia a docente, fuente autor	148
Ilustración 69. Vista frontend asignación de denuncia, fuente autor	149
Ilustración 70. Mockup Agregar preguntas al formulario, fuente autor	150
Ilustración 71. Mockup Agregar preguntas al formulario, fuente autor:	151
Ilustración 72.. Vista Frontend Agregar preguntas al cuestionario, Fuente Autor	153
Ilustración 73. Vista Frontend agregar preguntas al cuestionario, Fuente Autor	153
Ilustración 74. Vista Frontend timer, Fuente Autor	155
Ilustración 75. Agregar usuario, fuente autor	156
Ilustración 76. Vista Frontend agregar usuario, Fuente Autor	158
Ilustración 77. Diagrama de componentes, módulo docente, Fuente Autor	159
Ilustración 78. Cambio de contraseña, fuente autor	160
Ilustración 79. Vista Front log in - Fuente autor	161
Ilustración 80. mockup lista de casos, Fuente autor	162
Ilustración 81. Vista Front agregar nota, Fuente autor	164
Ilustración 82. Vista frontend de la gestión de denuncias, Fuente Autor	165
Ilustración 83. Prueba interfaz Administrador, Fuente autor	174
Ilustración 84. Prueba, Fuente autor	177
Ilustración 85.Resultados de análisis de pruebas de funcionalidad, Fuente autor	188
Ilustración 86. Pruebas funcionales. Fuente autor	209
Ilustración 87. Índice de severidad, fuente autor	213
Ilustración 88. Índice de severidad, Fuente autor	217
Ilustración 89, Índice de severidad, Fuente autor	221
Ilustración 90. Fragmento de código, Fuente autor	225
Ilustración 91. Registro docente, fuente autor	226
Ilustración 92. Índice de severidad, fuente autor	227
Ilustración 93, referenciarían, fuente autor	231
Ilustración 94, Mapa, Fuente, Google mapas	231
Ilustración 95, código, fuente autor	232
Ilustración 96, datos almacenados, fuente autor	233
Ilustración 97, índice de severidad de prueba, fuente autor	237
Ilustración 98. Código, fuente autor	237
Ilustración 99. Actualizar estado de la denuncia, fuente autor	238
Ilustración 100, código, fuente autor	239
Ilustración 101, Gestión de denuncia, fuente autor	239
Ilustración 102, estado de la denuncia, fuente autor	240
Ilustración 103. Agregar nota, fuente autor	241
Ilustración 104. notas de la denuncia, fuente autor	241
Ilustración 105. Cambiar contraseña, fuente autor	243

RESUMEN

En este trabajo se presenta el desarrollo de un prototipo de aplicación Web Móvil, para denuncias de Control Urbano de obras en construcción en la localidad 3 de la ciudad de Cartagena, “Gestor Urbano”, el cual hace parte integral de un Macroproyecto

perteneciente al Consultorio Jurídico de la Escuela de Derecho de la Universidad del Sinú, sede Cartagena.

La concepción de este prototipo tuvo como eje problemático el desplome del edificio Portal de Blas de Lezo II, acontecido en el mes de abril de 2017 en donde fallecieron 21 de las personas que trabajaban en la edificación [1], en vista de los fatales acontecimientos surge la idea de desarrollar una aplicación web móvil que permitirá al usuario identificar si las obras en construcción que están en determinada localidad, cumplen con los parámetros establecidos por el control urbano, esto crearía una medida de prevención para evitar irregularidades y posibles desgracias en el futuro. Este proyecto de desarrollo, creación e investigación es de gran importancia debido a que incentiva el uso de aplicaciones móviles que facilitan y permiten al ciudadano realizar procesos de consultas, quejas o reclamos y denuncias sobre el control urbano, en el Distrito de Cartagena de Indias.

Como objetivo general se planteó el desarrollo de un prototipo para una aplicación web móvil, que permita recibir y gestionar denuncias realizadas por la ciudadanía en general que impliquen control urbano en obras en construcción en la localidad tres de la ciudad de Cartagena.

Para iniciar la construcción de este prototipo, se analizaron los módulos a desarrollar en la aplicación, de acuerdo con las características solicitadas por el cliente (Consultorio Jurídico de la Universidad del Sinú, Sede Cartagena), a través de una serie de entrevistas a modo encuesta con los investigadores involucrados en el Macroproyecto, de donde se establecieron los requerimientos solicitados por el cliente, enfocándose en tres módulos: Administrador, Ciudadano y Docente.

Una vez realizado el análisis, se procedió con la etapa de diseño, en la cual se utilizaron Diagramas UML, los formatos establecidos en la Norma IEEE-830 y la realización del modelo conceptual mediante MOCKUP.

Finalizada la Fase de Diseño se dio inicio a la Codificación del Software utilizando el lenguaje de programación PHP, JavaScript y MySQL para el modelo de Base de Datos. En este proyecto se utilizó la metodología CMMI enfocada a procesos, como modelo de programación se empleó el Desarrollo Rápido de Aplicaciones RAD, el cual se

fundamenta en el desarrollo de software, el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering).

Concluida la fase de desarrollo se verificaron los requisitos del sistema y los requerimientos planteados por el cliente a través de un conjunto de pruebas, entre las que se tiene: pruebas de interfaz, funcionales y de severidad, las cuales fueron realizadas por ciclos y condujeron a resultados satisfactorios.

Se estima que la implementación de esta aplicación se lleve a cabo en el primer semestre del 2020, se proyecta a futuro realizar mejoras en el sistema al igual que ampliar algunas características y adicionar algunas nuevas, darle otros usos además de las denuncias de construcción y buscar que esta aplicación se pueda conectar con el Plan de Ordenamiento Territorial (POT).

INTRODUCCIÓN

En el presente proyecto se trabaja sobre una práctica que se ha vuelto común en el país y como referente de estudios nos enfocaremos en la ciudad de Cartagena, donde las construcciones ilegales se han tornado en actividades casi normales, según el periódico el Heraldo en el 2017 se hallaron 2.000 construcciones ilegales en la heroica, situación que termino en una desgracia que cobró la vida de 21 personas [1]. Estos hechos causan mucho temor en las comunidades y desestimula la inversión y el desarrollo inmobiliario de la Región, pero muy a pesar de esto, las denuncias de construcciones ilegales son muy pocas, esto es por diversos motivos como lo son, desconocimiento sobre la normativa, no saber cuándo una construcción es ilegal, evitar los largos trámites o tener que trasladarse desde un lugar a otro, ignorar donde o como interponer esta denuncia e incluso temor a denunciar; teniendo en cuenta esto, en este proyecto se presenta un alternativa que busca prevenir y crear pertinentes denuncias sobre las construcciones que no cumplan los parámetros de la ley, mediante un prototipo de aplicación web móvil, que permite al ciudadano contar con una herramienta útil, práctica, fácil de manejar y de agilidad.

En este proyecto se plantea el desarrollo de un software dirigido a colaborar en la problemática anteriormente enunciada y que afecta a la comunidad de Cartagena, se establecen objetivos específicos y generales que conducen a obtener los resultados esperados, se evidencian algunos proyectos similares, se describe la importancia de este software, los procesos y tipos de programación empleada y se enuncia la metodología usada para dicho fin.

1. DISEÑO METODOLÓGICO

1.1 Planteamiento del problema

1.1.1 Descripción del problema

El presente trabajo de grado se basa en una problemática actual que afecta a la comunidad Cartagenera, como lo es la carencia de denuncias ciudadanas en lo que concierne a construcciones ilegales o sin licencia, está claro que en determinadas situaciones es tedioso para el ciudadano trasladarse al lugar de las distintas entidades donde se puede presentar una denuncia, en algunas ocasiones la ausencia de las denuncias son producto del miedo, por evitar los largos trámites que esto conlleva o por considerarlas como una pérdida de tiempo; teniendo presente lo anterior, nace la necesidad de diseñar una alternativa para el ciudadano. Las personas como seres humanos, son seres cambiantes, evolucionan, cada día crean nuevas alternativas según las necesidades, quieren aprovechar mejor su tiempo, buscan la manera de emplear la tecnología como herramienta para hacer más fácil su día a día y ante estos cambios siempre encuentra una adaptabilidad; llevando a que en la actualidad predomine el avance y la tecnología, por lo cual muchos de los aspectos cotidianos de nuestras vidas son influenciados por estas dos palabras, en la actualidad que vivimos se crea la necesidad de desarrollar o implementar medios que faciliten las tareas, procedimientos, movimientos y demás actividades que hacemos o vemos diariamente. Las plataformas virtuales, aplicaciones móviles e internet se están convirtiendo en herramientas cada vez más indispensables en las Instituciones de Educación superior, en el ámbito laboral, aspectos generales y cotidianos de la vida.

La problemática de este proyecto se desarrolla en base a unas alarmantes cifras expuestas en el año 2017 sobre las construcciones ilegales y sus consecuencias en Cartagena, Un 84% de las construcciones visitadas por la administración distrital desde el 1 de mayo hasta el 6 de mayo de 2017, no contaban con los debidos permisos ni licencias. En total 50 inspecciones y 42 de ellas no tenían la documentación al día [1]

Es pertinente recordar que estas inspecciones surgieron por el terrible desplome del edificio Portal de Blas de Lezo II, la última semana del mes de abril de 2017 en donde fallecieron 21 de los obreros que trabajaban en la edificación.

En la Localidad 1, se visitaron 29 construcciones, de las cuales 25 no tienen licencia, en la Localidad 2 se visitaron 10 obras y todas están sin la respectiva licencia, por último, en la Localidad 3 se hicieron 11 visitas de las cuales 4 están sin licencia, 3 no llenan todos los requisitos y en las otras 4 no había nadie trabajando. Manuel Duque quien se encontraba ejerciendo el rol del alcalde durante este año reveló las cifras sobre la inspección a construcciones ilegales. Según la misma alcaldía, cerca del 80% de las mismas no cuentan con licencia, y si la tienen, presentan otro tipo de irregularidad que las hace estar en condición de ilegalidad. [2]

Aunque el porcentaje de obras sin licencias en 2017 fue y es alarmante, hay dos factores que empeoran la situación de las construcciones. Héctor Anaya que sucedió a Olimpo Vergara, después de las fallidas construcciones, como nuevo director administrativo de Control Urbano, explicó que, aunque en el reporte había proyectos que contaban con licencias legales de construcción, muchas de ellas violaban la normativa del POT. Es decir, en lugares en los que solo se podían construir hasta cuatro plantas, se llegaron a encontrar edificaciones que superan las seis. El edificio desplomado en Blas de Lezo, por ejemplo, supera el número de pisos permitidos en ese sector, lo que evidencia una falta de control gubernamental y veeduría ciudadana.

Otro requisito necesario para empezar las obras y que la mayoría de los proyectos no presentaron fue el plan de manejo ambiental, que es donde se establecen las acciones para prevenir, mitigar, controlar, compensar y corregir los posibles efectos o impactos ambientales negativos causados por un proyecto.

Es evidente que el desplome del edificio en Blas de Lezo generó mucho furor en el año 2017 y que las entidades encargadas estuvieron atentas a los procesos de muchas construcciones, pero actualmente en Cartagena cada día hay una construcción nueva y se entiende que es muy difícil estar en cada proceso de ellas, por lo cual la alternativa tecnológica que se presenta en este proyecto, busca evitar posibles incidentes y también dar a conocer la normativa y los mecanismos de acción para estos casos [1]. Esta aplicación para dispositivos móviles permitirá al usuario identificar si las obras que están en su barrio cumplen con los parámetros establecidos por el control urbano, esto crearía una medida de prevención ante posibles derrumbes o irregularidades.

El 20 de abril de 2018 la Fiscalía halló 20 edificaciones que carecían de licencias de construcción y cuyos reglamentos de copropiedad se hicieron con licencias falsas, el fin inmediato de esta aplicación es proporcionar un anonimato y un mecanismo de denuncias, discreto y de fácil uso para el usuario, el denunciar estas obras ilegales, agilizaría los procesos por parte del control urbano y podría salvar vidas [3].

1.1 Justificación

En la actualidad el uso de aplicaciones, que nos ayudan a facilitar trámites, que ahorran tiempo y en algunas ocasiones dinero, es muy común, como ejemplo se encuentran las aplicaciones de algunos bancos, centros comerciales y algunos mercados, estas entidades mediante una aplicación móvil, hacen que el cliente pueda realizar lo que usualmente le tomaba horas en solo en un clic desde la comodidad de su casa o cualquier lugar donde se encuentre, esta misma accesibilidad y comodidad es lo que se busca al desarrollar esta aplicación, facilitar los trámites denuncias, quejas, y

reclamos para acudir a las instancias legales tendientes a frenar y mitigar el impacto negativo de tales situaciones en el distrito.

Este proyecto de desarrollo, creación e investigación es de gran importancia debido a que incentiva el uso de aplicaciones móviles que permiten al ciudadano realizar procesos de consultas, casos de quejas o reclamos y denuncias sobre el control urbano, en el Distrito de Cartagena de Indias.

Muchos de los ciudadanos se benefician con esta aplicación dado que se ahorrarán los desplazamientos para ubicar la correspondiente dependencia donde establecer la denuncia, los largos trámites y el tiempo de espera usual para recibir la atención necesaria, y lo más importante es que con el uso de esta aplicación también se podría evitar accidentes causados por las faltas de denuncias ante estas construcciones, o cualquier anomalía que se puede presentar.

Ahora bien, como aporte para dar solución a estos impases, se nos permitió crear mediante los espacios virtuales, el diseño y desarrollo de un prototipo para la diligencia de denuncias de control urbano, bajo una aplicación web móvil, en la que se podrá gestionar casos de quejas o reclamos, denuncias sobre el control urbano, donde el usuario/ciudadano, después de ingresar sus datos, este podrá encontrar distintas alternativas dentro de la aplicación, además el usuario podrá tomar evidencias fotográficas de la edificación que no cumpla los parámetros de construcción y este será enviado al administrador quien le responderá al denunciante y hará el ingreso del caso para tomar las medidas que sean necesarias, el usuario también contará con la opción de enviar la ubicación y la opción de consultar sobre las normas y parámetros de ley establecidos. Este aspecto hace que este proyecto sea viable y que aporte a la sociedad mecanismos de participación e interacción legal.

1.1 Formulación del problema

¿Cómo facilitar a la ciudadanía realizar procesos de denuncia sobre el control urbano de obras en construcción en la localidad tres de Cartagena?

1.2 Objetivos

Objetivo general

Desarrollar el prototipo de una aplicación web móvil, que permita recibir y gestionar denuncias de control urbano de obras en construcción, realizadas por la ciudadanía en la localidad tres de la ciudad de Cartagena.

Objetivos específicos

- Analizar los módulos a desarrollar en la aplicación, de acuerdo con las características solicitadas por el cliente. (Consultorio Jurídico de la Universidad del Sinú, Sede Cartagena)
- Diseñar mediante diagramas UML la información descrita en los requerimientos del sistema.
- Desarrollar un prototipo de App en lenguaje de programación PHP para denuncias de control urbano de obras en construcción, de acuerdo con los diseños planteados.
- Realizar el conjunto de pruebas que permitan validar los requerimientos y requisitos del sistema, planteados por el usuario aplicándolos en la localidad tres de Cartagena.

1.3 Alcance del proyecto

Este proyecto se centra en la realización de un prototipo de App web móvil basado en una necesidad del consultorio jurídico del programa de derecho de la universidad del Sinú seccional Cartagena, en el cual se pretende recibir y gestionar denuncias de control urbano de obras en construcción realizadas por la ciudadanía. Este proyecto consta de tres módulos principales (Ciudadano, Administrador y Docente) y cada uno de estos cuenta con sus respectivos permisos de usuario.

El módulo del ciudadano consiste en permitir al usuario registrarse en la aplicación, la cual le dará acceso para identificar la localización de la obra en construcción que se presume esté infringiendo los requisitos mínimos de Ley para su construcción; estas características de obra en construcción se encuentran en un formulario diseñado en la aplicación y el ciudadano deberá ir dando respuesta a cada una de las preguntas, luego de eso, este deberá adjuntar las imágenes como soporte del caso, la aplicación tomará y dará de referencia la ubicación de la denuncia. Una vez realizado esto, las denuncias serán recibidas por un administrador en consultorio jurídico y a su vez, este asignará a un docente para su respectiva gestión en campo.

El docente deberá realizar los trámites necesarios y dar respuesta al ciudadano a través del módulo. A medida que esto ocurre, el ciudadano podrá consultar el estado de su denuncia al igual que monitorear el proceso de esta. A continuación, se describen los involucrados relevantes del proyecto.

Tabla 1. Involucrados Relevantes

N. o	Involucrado	Tipo	Encargado de	Compromisos
	Involucrados relevantes	Interno/Externo al proyecto	Qué rol tiene dentro del proyecto	Qué actividades tiene a su cargo
1	Deimer Ávila Argote	Interno	Líder de Proyecto, Documentación	Levantamiento de requerimientos, documentación de código, desarrollo de interfaz gráfica para los módulos.
2	Brayan Bravo Silva	Interno	Desarrollo	Desarrollo de interfaz gráfica para los módulos: Ciudadano, Administrador y Docente. Realizar conexión a la base de datos.
3	Luis Blanquicet Benavidez	Interno	Asesor Disciplinar	Asesoría durante el desarrollo del proyecto.
4	Eugenia Arrieta	Interno	Asesora Metodológica	Asesoría en el la parte metodología del proyecto y revisión literaria.
5	Alonso Cortina Acevedo	Externo	Cliente	Validación Aprobación de requerimientos
6	Vanessa Quintana	Externo	Cliente	Validación Aprobación de requerimientos
7	Yosiris Sierra Brango	Externo	Cliente	Validación Aprobación de requerimientos

1.4 Estado del arte

En esta etapa se permite conocer otras investigaciones que aportan una clarificación de ideas respecto al tema de interés, y así podrá definirlo mejor, afinarlo, delimitarlo, y enfocarlo desde la perspectiva que interesa, en este caso será el enfoque sé que maneja durante el proyecto es tecnológico y legal. A continuación, se describirán algunas aplicaciones móviles que se iniciaron como proyectos de grado y fueron de utilidad a la comunidad: En primera instancia, se describe el aporte que hace a la investigación y las similitudes encontradas, analizando su naturaleza e identificando características relevantes de su desarrollo y el diseño de estas aplicaciones móviles de alerta o de gestión.

“CuidalApp” es una aplicación web móvil que permite entre otras cosas, reportar incidentes y problemas urbanos de diferente índole, recopilando información importante para su gestión y solución, tales como datos geográficos y de geolocalización, tipología, nivel de riesgo, imágenes, videos y descripción detallada del incidente, esto con el fin ser asignado automáticamente a un equipo de trabajo especializado (docentes y estudiantes de Derecho) o multidisciplinar (líderes comunales y ciudadanos cercanos al lugar del incidente), quienes deben realizar y reportar las diligencias pertinentes y necesarias para resolver el problema o incidente Urbano, guardando en el sistema cada evidencia (documental, fotográfica, videos, conversaciones, etc.) que permita hacer seguimiento y trazabilidad sobre la gestión de dicha solución. CuidalApp tiene muchas similitudes con este proyecto, como primera similitud encontramos que esta aplicación móvil hace parte integral de un Macroproyecto perteneciente al Consultorio Jurídico de la Escuela de Derecho de la Universidad del Sinú, sede Cartagena. También como antes se mencionó esta plataforma permite reportar sobre incidentes y problemas siendo un punto en común el área urbana, la diferencia entre ambos proyectos yace en que “Gestor Urbano” apunta directamente a obras en construcción y el cumplimiento de

las normas según el plan de ordenamiento territorial, mientras que “CiudadApp” apunta a la gestión de incidencias tales como: Incidentes viales(accidentes, tráfico, congestión de vías), alumbrado público, sistema de alcantarillado y aseo público, problemas con las zonas verdes y problemas con la seguridad y delincuencia. esta plataforma constituyo un gran referente en el desarrollo de este proyecto dado que hace parte de una integración a futuro como un módulo anexo del macroproyecto [4].

Otra aplicación que sirvió como referente fue “El diseño y desarrollo de una aplicación móvil para dispositivos Android con un sistema de alerta temprana para prevención de desbordamientos de arroyos de la ciudad de barranquilla”. como primera característica a destacar de este software es, la gran afinidad al desarrollado en este proyecto, dado que en este se presenta un proyecto de grado que buscaba la elaboración de una aplicación móvil para el sistema operativo Android utilizando la tecnología del Api de Google Maps, bajo el concepto del patrón de diseño de arquitectura de software MVC (Modelo – Vista -Controlador) y estructurada con la metodología de programación en cascada, la cual se soporta en fases de desarrollo como la definición de los requisitos, el diseño del software, implementación de pruebas unitarias, integración, de pruebas del sistema, operación y mantenimiento [4]. Ahora bien, esta aplicación al igual que “Gestión de Licencias Urbanísticas” tiene como propósito el prevenir ciertas catástrofes que en determinado caso se pueda presentar y afectar la comunidad, por esto encontramos pertinente seleccionar este trabajo de grado dado que también busca prevenir y su carácter representa un beneficio social, manejando un sistema de alerta, aunque con distintas modificaciones y diseño, puede aportar mucho a nuestro proyecto “Gestión de Licencias Urbanísticas”.

Como referente también se revisó un proyecto que comparte el sistema de alerta similar al nuestro, “aplicación móvil Android, para la sistematización del estudio, la prevención y el control de enfermedades en los cultivos de algodón como herramienta de apoyo para el aprendizaje de los estudiantes del programa ingeniería agronómica de la universidad de córdoba, este trabajo de grado es sobre la aplicación móvil IPC

algodón” [5], el cual constituye una importante innovación en el área del estudio, la prevención y el control de las enfermedades en el cultivo de algodón, y llega a ser de gran ayuda pedagógica tanto para los docentes del área como para los estudiantes del programa de ingeniería agronómica de la Universidad de Córdoba, se escogió este proyecto de grado porque aplica dos aspectos similares con el proyecto en curso, el primero es el desarrollo de una aplicación móvil que busca la prevención y como segunda instancia es que se unen dos programas académicos, en este caso la fusión está dada entre la ingeniería en sistema y el programa de derecho.

Adicionalmente este proyecto tiene como característica principal la automatización de las denuncias, que como se mencionaba en el planteamiento, requieren de un trámite dispendioso, mucho tiempo para realizarla y usualmente no se puede presentar de forma completamente anónima, por lo cual una de las funciones de nuestro proyecto es que estos trámites se conviertan en procesos ágiles y automatizados, por lo cual es pertinente tomar como referencia la tesis previa a la obtención del título de ingeniero de sistemas, titulada “Desarrollo de una aplicación, para dispositivos móviles que permita administrar pedidos y controlar rutas de los vendedores, aplicada a la empresa “almacenes Juan Eljuri Cía. Ltda. división perfumería”, este [6] busca automatizar procesos en la gestión de pedidos que se viene realizando de forma manual, es importante como referencia debido a que esta busca agilizar y automatizar los procesos de gestión y lograr que esta aplicación sea más práctica para los ciudadanos en el proceso de denuncias para el control urbanístico de manera fácil y automática.

1.5 Marco referencial

Marco teórico

Es evidente que se está ante un mundo donde el impacto de las tecnologías de la información y la comunicación afecta a todos los ámbitos de nuestra vida, vivimos en un tiempo de cambios que trascurren a una velocidad vertiginosa. La ciencia y la

tecnología generan grandes cambios e innovaciones que se manifiestan inevitablemente en nuestro entorno.

Las tecnologías digitales representadas por creaciones multimedia interactivas y su unión a las redes constituyen un eje muy importante en la revolución subyacente que está impregnando el mundo actual. La singular combinación de textos, imagen y sonido, abren las posibilidades de participación del usuario. El universo multimedia interactivo en red presenta un gran potencial técnico para el desarrollo de aplicaciones y recursos pedagógicos.

El autor Nicholas Carr plantea que nuestros cerebros se adaptan al ambiente, y ciertamente la tecnología de los computadores hace parte importante de ese ambiente hoy. Nos apoyamos en los computadores para hacer cada vez más actividades diarias. Como lo han mostrado ciertos estudios, la mejor manera de aprender es lidiar con los retos difíciles, pues es en medio de la dificultad donde los talentos se expanden [7]

Ciertamente la evolución de las redes telemáticas nos ofrece la oportunidad de poder trabajar con una gran variedad de personas, a veces lejanas a nivel espaciotemporal. Este entorno tecnológico plantea realizar un tratamiento óptimo y eficaz de una gran cantidad de información que Internet proporciona. Así mismo, demanda establecer nuevos criterios de elaboración y tratamiento de dicha información y crear nuevas pautas de comunicación e interacción de cara al siglo XXI.

Este nuevo panorama sociocultural representa importantes implicaciones pedagógicas. Las personas necesitan las nuevas habilidades requeridas en los ámbitos profesionales. Así, más que nunca, las personas necesitan: desarrollar autonomía e iniciativas propias para emprender y auto dirigir procesos de aprendizaje personales, comunicar ideas con claridad y precisión por medio de variedad de recursos técnicos, agilizar la toma de decisiones, tener disposición y voluntad para formarse y trabajar de manera cooperativa. Por otra parte, es importante destacar el tipo de programación que

se utiliza para el desarrollo de esta aplicación móvil, a continuación, se encuentra el concepto y uso de este.

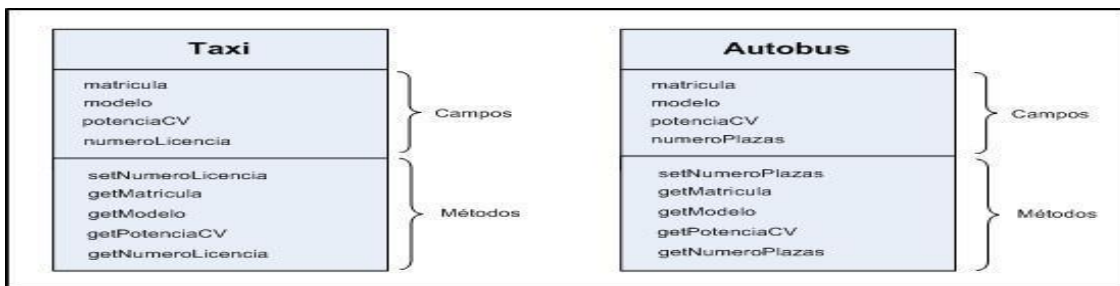
Programación Orientada a objetos

(POO) es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real que otros tipos de programación. Con la POO tenemos que aprender a pensar las cosas de una manera distinta, para escribir nuestros programas en términos de objetos, propiedades, métodos y otras cosas que veremos rápidamente para aclarar conceptos y dar una pequeña base que permita soltarnos un poco con este tipo de programación. Por ejemplo, en un proyecto que utilice objetos Taxi y objetos Autobús, como se presenta en la Ilustración 1:

Ilustración . POO, fuente code.google.com

MVC (Modelo. Vista. Controlador)

Es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz que ve el usuario de la lógica del negocio en tres componentes distintos. Es más frecuente en aplicaciones Web que en aplicaciones de escritorio, sin embargo, es



aplicable también a este, sin ningún problema, Java ya contaba hace rato con Observar y Observable, herramientas que nos ayudan a la interacción entre la interfaz y el modelo, sin embargo, el ejemplo que dejamos a continuación no hace uso de estas herramientas [8].

La descripción del patrón MVC es: Vista (View):

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. [9]

Ilustración . Código manual PHP, fuente Manual php

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo</title>
  </head>
  <body>

    <?php
      echo "¡Hola, soy un script de PHP!";
    ?>

  </body>
</html>
```

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!").

El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML

con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga. [9]

MySQL

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

Se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP). [10]

Framework

Un Framework, que se podría traducir aproximadamente como marco de trabajo, es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado. Esta definición, algo compleja, podría resumirse como el entorno pensado para hacer más sencilla la programación de cualquier aplicación o herramienta actual.

Este sistema plantea varias ventajas para los programadores, ya que automatiza muchos procesos y además facilita el conjunto de la programación. Es útil, por ejemplo, para evitar el tener que repetir código para realizar funciones habituales en un rango de

herramientas, como puede ser el acceder a bases de datos o realizar llamadas a Internet. Todas estas tareas son las que se realizan de forma mucho más fácil cuando se trabaja dentro de un Framework. [11]

API

Las API son un conjunto de comandos, funciones y protocolos informáticos que permiten a los desarrolladores crear programas específicos para ciertos sistemas operativos. Las API simplifican en gran medida el trabajo de un creador de programas, ya que no tiene que «escribir» códigos desde cero. Estas permiten al informático usar funciones predefinidas para interactuar con el sistema operativo o con otro programa. [12]

Google Mapas

Google Mapas es el nombre de una aplicación desarrollada por Google que se encarga de ofrecer a los usuarios toda la información que necesiten sobre su ubicación actual, como también la de cualquier dirección específica, así como el trazado de recorridos para llegar al lugar que estos deseen desde donde se encuentran.

Una aplicación que aprovecha la conexión GPS de los smartphones, tables y la combina con el seguimiento de los mapas y los callejeros que realiza Google para dar a los consumidores una herramienta de lo más eficaz. Gracias a este software, no es necesario comprar mapas ni tampoco preguntar por direcciones, puedes buscarlas directamente en tu teléfono móvil y, además, saber por dónde ir para llegar en el menor tiempo posible. [13]

SQL (Structured Query Language)

Es un lenguaje estándar e interactivo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas, gracias a la utilización del álgebra y de cálculos relacionales, el SQL brinda la posibilidad de realizar consultas con el objetivo de recuperar información de las bases de datos de manera sencilla. Las

consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar, averiguar la ubicación de los datos, y más.

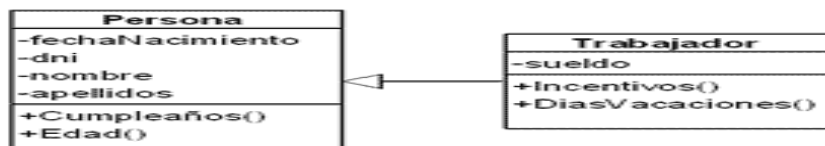
Lenguaje unificado de modelado (UML)

Fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación [14].

Ilustración . Ejemplo UML, Fuente code.google.com



Ejemplos de símbolos UML usados en el Diagrama de Casos de Uso



Ejemplos de símbolos UML usados en el Diagrama de Clases

Marco conceptual

- Consultas: vinculado a solicitar el parecer o la instrucción de otra persona. Quien consulta espera obtener información de utilidad para satisfacer sus necesidades o conseguir sus objetivos [15]
- Aplicación móvil: aplicación de software que se instala en dispositivos móviles o tabletas para ayudar al usuario en una labor concreta, ya sea de carácter profesional o de ocio y entretenimiento, a diferencia de una web App que no es instalable [16].

- Tecnología: ciencia que se aplica para dar solución a los problemas que se presentan en la sociedad, en diferentes sectores y se utiliza para desarrollar alternativas que puedan ayudar a que las personas lleguen a tener una mejor vida en todo tipo de aspectos [17].
- Denuncias: poner en conocimiento por parte de un consumidor ante una Autoridad Administrativa de unos hechos que sustentan una posible vulneración de sus derechos, así como los derechos de otros consumidores [18].
- Plataformas virtuales: un entorno informático en el que nos encontramos con muchas herramientas agrupadas y optimizadas para fines docentes. Su función es permitir la creación y gestión de cursos completos para internet sin que sean necesarios conocimientos profundos de programación [19].

Marco legal

Constitución Política de Colombia.

- Ley 1712 de 2014 – De Transparencia y Acceso a la Información pública.
- Decreto 1151 de 2008, Decreto 2693 de 2012 Lineamientos generales de la Estrategia de Gobierno en línea.
- Ley 872 de 2003 Por la cual se crea el Sistema de Gestión de Calidad en la Rama Ejecutiva del Poder Público y en otras Entidades. prestadoras de Servicios

- Decreto 1122 de 1999. Ley estatutaria que se expida sobre el derecho de petición.
- Decreto Ley 019 de 2012. Ley 1437 de 2011. Ley 872 de 2003. Artículo 4. Ley 527 de 1999. Regulan el proceso de automatización de las entidades públicas.
- Norma ISO internacional 15489-1 - PGR Información y documentación – Gestión de documentos.
- Norma técnica de calidad NTCGP 1000:2009 (Calidad en la Gestión pública) MECI
- Normatividad/Política CONPES 3654 de 2010 "Política de rendición de cuentas de la Rama Ejecutiva a los ciudadanos".
- Ley 1454 de 2011, que dicta las normas orgánicas para la organización político-administrativa del territorio colombiano, fortaleciendo la identidad cultural de las regiones y el desarrollo territorial.

Selección del modelo y metodología

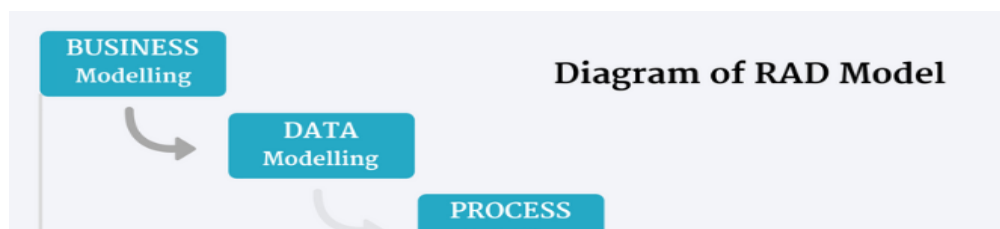
Rapid application Development (RAD)

En este proyecto se implementa el modelo de desarrollo rápido de aplicaciones RAD (acrónimo en inglés Rapid Application Development) este es un proceso de desarrollo de software, desarrollado inicialmente por James Martin en 1980. El método comprende el desarrollo interactivo, la construcción de prototipos y el uso de utilidades CASE

(Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

Fases de RAD

- Modelado de gestión: el flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la proceso?
- Modelado de datos: el flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.
- Modelado de proceso: los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos. Es la comunicación entre los objetos.
- Generación de aplicaciones: El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas automáticas para facilitar la construcción del software.
- Pruebas de entrega: Como el proceso DRA enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo. [20]



Ahora bien, como metodología para la ejecución de este proyecto, se escogió CMMI la cual contiene las mejores prácticas y provee a las organizaciones de aquellos elementos que son esenciales para que los procesos de negocio de estas sean efectivos [21]

El modelo CMMI fue inicialmente desarrollado para los procesos relativos al desarrollo e implementación de Software por la Carnegie-Mellon University. Este vio la luz por primera vez en el año 1987 como Capability Maturity Model CMM. Dicho nombre, tanto como los cinco niveles de la representación por etapas, están inspirados en el modelo de madurez Manufacturing Maturity Model de Crosby [22].

En principio el modelo CMM era aplicado en programas de defensa, pero lo cierto es que este ha logrado gran aceptación, tan es así que ha sido sometido a varias revisiones e iteraciones. Debido a su éxito se llevó a cabo el desarrollo de modelos CMM para para diversos ámbitos más allá del software [23].

Fases del Capability Maturity Model integración (CMMI)

Se divide en 5 pasos o niveles de desarrollo, dentro de cada uno de estos niveles se engloban una serie de objetivos claves para poder pasar al siguiente nivel. La visión que se le puede dar al modelo de CMMI puede ser doble, por un lado, encontramos la representación continua y por otro la escalonada.

La versión continua hace un desglose por nivel de proceso dejando a un lado la parte de madurez, presenta un nivel más, en total 6, repartidos de la siguiente manera [24] :

- Nivel 0 o incompleto
- Nivel 1 o realizado

- Nivel 2 o gestionado
- Nivel 3 o definido
- Nivel 4 o gestionado cuantitativamente
- Nivel 5 u optimizado

Y la versión escalonada se divide al igual que antes en 5 etapas que a medida que ascendemos tienen un mayor grado de desarrollo, los niveles de madurez serían:

- Nivel 1 o inicial, en este nivel se encuentran empresas que abarcan demasiados objetivos incumplidos, falta de desarrollo de procesos o que directamente no se ven capacidad de desarrollar las nuevas ideas.
- Nivel 2 o gestionado, aquí encontramos empresas que tienen sus proyectos desarrollados están bien definidos, planificados tienen un buen seguimiento de kpi's y tienen asegurados un control.
- Nivel 3 o definido, una vez llegados a este nivel podemos ver empresas que desarrollan sus proyectos mediante estándares, herramientas, procedimientos...es decir, vemos que los proyectos se siguen de una forma establecida según la organización, pero de una forma clara y específica.
- Nivel 4 o gestionado cuantitativamente, se busca el establecimiento de objetivos cuantitativos y ejecución de estos, con ello se consigue tener en cuenta las necesidades de los clientes, y cómo saldrá el producto para el cliente final. En esta etapa se miden los procesos con datos, por lo que se realizan cálculos estadísticos.

- Nivel 5 u optimizado, con el análisis de los datos se puede comprender dónde detectar la mejora y actuar al respecto, y conseguir una mayor eficiencia del proceso. La ilustración número 5 describe las fases de la metodología:

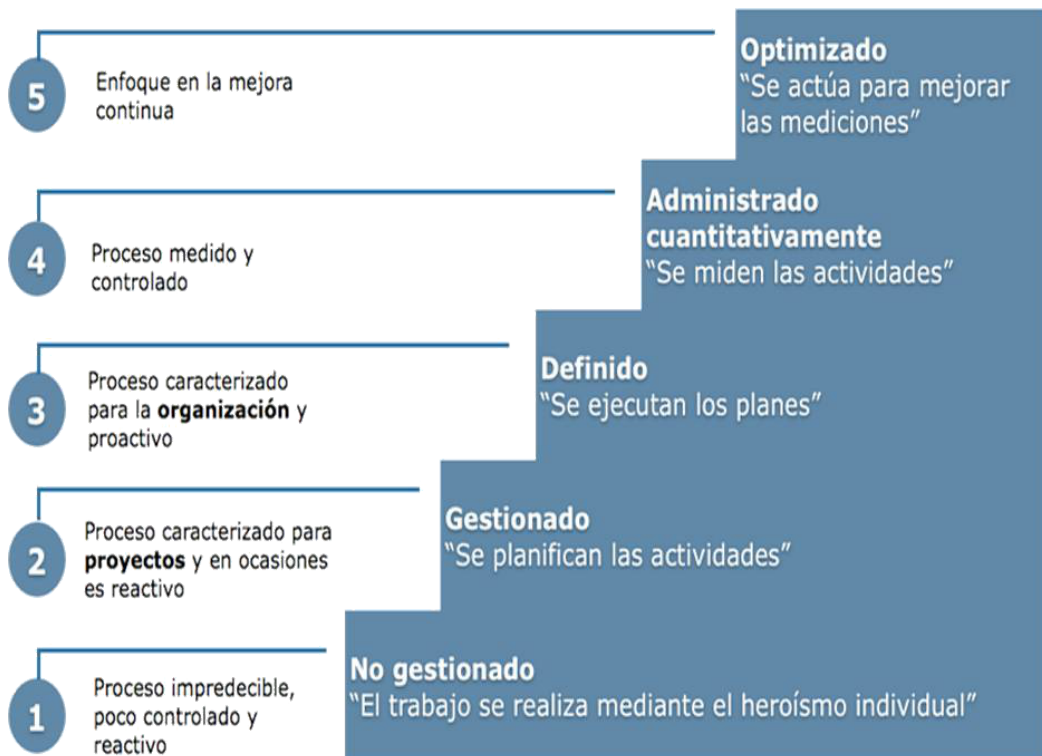


Ilustración 5 Fases del capability maturity model integration (cmmi), Fuente autor.

De igual modo si se utiliza una u otra metodología del modelo CMMI hay que ser consecuentes en que hay que pasar un reconocimiento para cualquiera de los niveles mediante SCAMPI (Standard CMMI Appraisal Method Process Improvement), para esta tarea es necesario de un equipo preparado y cualificado. Este tipo de evaluación también sirve para detectar cuáles son las fortalezas y las debilidades en los procesos sobre los que se trabaja e identificar posibles riesgos [22].

2. ANÁLISIS DEL SISTEMA

En este capítulo se detalla el levantamiento de los requerimientos del sistema basado en una serie de entrevistas con el cliente, las cuales facilitaron la descripción específica de cada uno de los componentes del sistema, así como los módulos a desarrollar y cada una de sus características.


2.1 Alcance del proyecto

El proyecto en desarrollo se identifica con el nombre **Gestor Urbano**, este se caracteriza por ser una aplicación funcional e intuitiva al ciudadano para la realización de denuncias, por medio de geolocalización, lo cual permitirá la fácil identificación de infracciones e irregularidades en obras en construcción.

Método para recolección de la información

Para el levantamiento de esta información, fue seleccionada la entrevista como medio de interacción directa con los involucrados relevantes del proyecto, por cada una de las

entrevistas se realizan las respectivas actas o minutas para llevar el control de las visitas realizadas, de esto modo, poder tener una perspectiva a detalle de las características del sistema, además de tener en claro cuáles serían las personas al frente del proyecto. A continuación, se presenta el formato de acta utilizado para cada entrevista:

 UNIVERSIDAD DEL SINÚ Elías Bechara Zainúm Seccional Cartagena	PROCESO: GESTIÓN DIRECTIVA
	TÍTULO: ACTA DE REUNIÓN
	CÓDIGO: R-GDI-001
	VERSIÓN: 001

FECHA DE REALIZACIÓN	03/07/2018	FECHA PROGRAMADA	29/07/2019	ACTA No.	
----------------------	------------	------------------	------------	----------	--

LUGAR DE LA REUNIÓN:	SEDE PLAZA COLÓN
----------------------	------------------

HORA INICIO:	3:00 PM	HORA FINALIZACION:	5:30 PM
--------------	---------	--------------------	---------

OBJETO DE LA REUNIÓN	PROYECTO "APP Control Urbano"
----------------------	-------------------------------

TIPO DE REUNIÓN : ORDINARIA EXTRAORDINARIA

1. ASISTENTES

No.	NOMBRE COMPLETO	CARGO	FIRMA
	Alonso Cortina Acevedo	Docente Investigador	En Original
	Vanessa Quintana	Coordinadora de Investigación	En Original
	Yosiris Sierra Brango	Estudiante de Semillero Ius Gentium	En Original

Ilustración 5. Instrumento de recolección de datos, Fuente autor.

En el levantamiento inicial de la información se llevó a cabo una serie de preguntas las cuales fueron formuladas a los siguientes involucrados en el proyecto:

- Alonso Cortina Acevedo (Docente Investigador)
- Vanessa Quintana (Coordinadora de Investigación)
- Yorisirs Sierra Brango (Estudiante de derecho)

Los criterios que se tuvieron en cuenta para la obtención de requerimiento y levantamiento del alcance del producto fueron los siguientes:

1. ¿En qué consiste el proyecto?
2. ¿Cuáles son las actividades que se involucran en el proceso?
3. ¿Quiénes son los actores que intervienen en el proceso?
4. ¿Cuáles son las funciones de cada proceso?
5. ¿Qué datos utiliza o produce este proceso?
6. ¿Qué validaciones se deben realizar sobre estos datos?
7. ¿Qué eventos pueden ocurrir durante la ejecución del proceso?
8. ¿Qué decisiones pueden tomarse durante la ejecución del proceso?
9. ¿Quiénes toman decisiones con la información resultante?
10. ¿Restricciones deberían existir en el sistema?
- 11.

Teniendo en cuenta lo anterior se procedió a realizar el detalle de la información recopilada la cual dio lugar a las siguientes especificaciones requeridas por el cliente:

Realización de la denuncia (ciudadano)

- ✓ Inicio de sesión del ciudadano: usuario ciudadano deberá autenticarse en la aplicación con los siguientes datos mínimos: nombre, correo electrónico, teléfono.

- ✓ La aplicación permitirá al ciudadano mediante un formato prediseñado, diligenciar la información que luego será generada automáticamente por la aplicación, esta será la denuncia; en dicha denuncia, el ciudadano podrá decidir si quiere que el consultorio jurídico de la universidad del Sinú realice el trámite o si él mismo quiere acercarse a la estación más cercana para realizar dicho proceso.
- ✓ El ciudadano podrá consultar las incidencias que haya realizado a través de la aplicación, ya sea por el número de radicado o por rango de fecha de creación.

Gestión de la denuncia (caso 1)

- ✓ En caso de que el ciudadano decida realizar la denuncia el mismo, la aplicación le enviará un correo electrónico (Formato de denuncia) en el cual se especificará el número de radicado con la estación de policía a la cual este se debe dirigir.
- ✓ La aplicación deberá enviar la denuncia por medio de correo electrónico informando a consultorio jurídico como constancia de que el ciudadano realiza el proceso por sí mismo

Gestión de la denuncia (caso 2)

En el caso de que el ciudadano decida que el consultorio jurídico sea quien gestione la denuncia, la aplicación contará con un módulo de administración para la gestión de cada caso. Cada denuncia contará con un número de radicado el cual podrá ser asignado al personal correspondiente por el administrador del módulo. El administrador del módulo para la gestión de los casos podrá realizar las siguientes acciones:

- ✓ Inicio de sesión en la aplicación: el usuario administrador deberá autenticarse para poder realizar las respectivas actividades en el sistema.

- ✓ Visualizar los casos generados por el ciudadano; fecha de creación de la incidencia, lugar, hora y número de radicado de la incidencia.
- ✓ Asignar incidencias al secretario asignado: Visualizar lista de secretarios a los cuales les puede asignar el caso.
- ✓ Visualizar las estadísticas de incidencias: el administrador del módulo de gestión podrá visualizar los reportes de incidencias en Excel o PDF con la información detallada de cada caso como: consecutivo de la denuncia, usuario quien genera la denuncia, secretario asignado la para la gestión de la denuncia, fecha de realización de la denuncia, hora de la realización de la denuncia, clasificación de la denuncia. Descripción detallada de la denuncia según los datos ingresados por el ciudadano dado por el resultado del formato de denuncia.
- ✓ El administrador del módulo de gestión de denuncias podrá realizar modificaciones de los cuestionarios requeridos según sean las actualizaciones de los reglamentos establecidos por el ente territorial.

Seguimiento de la denuncia (ciudadano)

El ciudadano recibirá notificaciones a través de correo electrónico, acerca de la denuncia realizada, la cual sea tramitada por el consultorio jurídico de la universidad del Sinú

Los actores con rol de Administrador, secretario (docente), o a quien corresponda por parte del consultorio jurídico, podrán agregar notas como estado de la denuncia, actualización en cuanto a seguimiento, a las denuncias realizadas por

el usuario ciudadano. El usuario ciudadano podrá consultar a través del código de la denuncia, el estado de las solicitudes realizadas.

Criterios para la realización de la denuncia

El usuario (Ciudadano) encontrará la siguiente información como requisitos mínimos al momento de realizar la denuncia:

1. ¿Tiene la obra instalada la valla informativa de la licencia de construcción?
2. ¿La obra se construye sobre espacio público?
3. ¿La obra afecta la integridad del espacio público? (andén – calle)
4. ¿La obra ocupa con material de construcción el andén, las vías o espacios públicos circundantes?
5. ¿Representa la obra un peligro para el transeúnte?
6. ¿Cuenta la obra con la debida protección para evitar la contaminación del sector donde se ubica? (Se refiere este punto a mallas o barreras de protección)
7. ¿Desarrolla la obra actividades de construcción en la vía pública, afectando el entorno? (Eje. Preparación de mezclas en la vía pública)

Estas preguntas serán comparadas por el algoritmo de la aplicación y le indicarán al usuario si hay o no infracción generando automáticamente la denuncia según sea el caso.

Características de la aplicación

La aplicación contará con las siguientes características:

- ✓ Estandarización de las incidencias y su respectiva gestión.
- ✓ Visualización de las incidencias más recientes.
- ✓ Visualización de las zonas con mayor participación de incidencias.

- ✓ Visualización de las incidencias más comunes y frecuentes.
- ✓ Se podrá consultar la Visión estratégica
- ✓ Se podrá consultar las preguntas frecuentes
- ✓ Se podrá consultar la Instrucción de manejo de la aplicación

Perspectiva del producto

La App web móvil Gestor Urbano, es una aplicación que se encarga de generar y gestionar las denuncias realizadas por el ciudadano en relación con las obras en construcción.

Es importante resaltar, que no se conoce una aplicación de esta categoría, ya que nos brindará la facilidad de gestionar y agilizar el proceso de denuncias de control urbano, permitiendo así el diseño de un sistema innovador en esta área.

Funciones del proyecto

El desarrollo de la App gestor Urbano cuenta con las siguientes funciones:

- Administrar la información de usuarios (ciudadanos), denuncias.
- Gestionar la información registro de denuncias (Formato para el reporte de la incidencia): Registro de pruebas, marcación geográfica de la incidencia, envío de registro de pruebas y formato, Recepción de la incidencia a consultorio jurídico por medio de correo electrónico.
- Gestionar la asignación de las denuncias: grupos de trabajo (consultorio jurídico).
- Gestionar herramientas de apoyo: GOOGLE MAPS.

- Desarrollo de interfaces para fácil manejo de denuncias: secretario, grupo de trabajo del consultorio jurídico.

La tabla número 2 muestra con un poco más de detalle las funciones del proyecto descritas en el alcance:

Tabla 2. Funciones del proyecto, Fuente Autor

Actores y características de sistema	FUNCIONES
Ciudadano	<ul style="list-style-type: none"> ● Gestionar usuario ● Gestionar la incidencia ● Autenticación de usuario ● Seguimiento de las denuncias registradas
Administrador	<ul style="list-style-type: none"> ● Gestionar la asignación de todas las denuncias registradas. ● Visualizar herramienta informativa (Estandarización de las incidencias y su respectiva gestión, zonificación de las incidencias más recientes, las zonas con mayor participación de incidencias) ● Gestión de reportes. ● Gestión de usuarios.

Docentes	<ul style="list-style-type: none">● Gestionar los casos asignados por el administrador para dar inicio al proceso denunciado.● Visualizar herramienta informativa (Estandarización de las incidencias y su respectiva gestión, zonificación de las incidencias más recientes, las zonas con mayor participación de incidencias).
Adaptabilidad	<ul style="list-style-type: none">● Dar soporte de actualización a las normativas o artículos de cada incidencia.

Características del usuario.

Las características de los usuarios se expresan a través de la siguiente jerarquía:



Jerarquía usuarios del sistema: Los usuarios potenciales del sistema son todos los ciudadanos, administrador, profesores y estudiantes (secretarios) del consultorio jurídico.

Restricciones generales

Se tendrá en cuenta el modelo actual de la dependencia de control urbanístico de la ciudad de Cartagena, el cual presenta constantes falencias en su estructura de reporte de incidencias.

Este proyecto estará sujeto al tiempo de desarrollo contemplado durante dos meses y medio (marzo, abril y mayo) bajo la supervisión de los docentes de proyecto metodológico e interdisciplinar y teniendo en cuenta del cronograma de trabajo.

Suposiciones y dependencias

Algunos factores que pueden afectar los requerimientos del sistema son:

- Modificación de la aplicación.
- Modificación de normativas.
- Agregar nuevas funcionalidades a las ya definidas anteriormente.
- El lenguaje de programación a utilizar no apto para implementar los requerimientos definidos.
- Falta de entrega o retraso de los insumos como soportes, formatos, políticas a incluir en la aplicación, solicitados al cliente para la configuración en el sistema.

Requisitos del sistema

Los requisitos del sistema fueron establecidos utilizando el formato presentado en la tabla número 3 donde se detalla los siguientes datos: función, descripción, entradas, fuentes, salidas, proceso, restricciones, precondiciones, post condiciones, contacto y fecha.

Tabla 3. Requerimientos. Fuente Autor

RF 000. Entidad-Número Requerimiento	
Función	<i>Nombre del requerimiento</i>
Descripción	<i>Se describen las características del requerimiento</i>
Entradas	<i>Datos para registrar según la funcionalidad del requerimiento</i>
Fuentes	<i>Refiere a la fuente donde se obtienen dichos datos</i>
Salidas	<i>Es el resultado esperado, a modo de confirmación visible al usuario</i>
Proceso	<i>Se describen las actividades realizadas para la captura de datos</i>
Restricciones	<i>Se detallan los procesos excepcionales en caso de que las condiciones de ingreso de datos no sean según los datos solicitados.</i>
Precondiciones	<i>Estado en el cual el sistema debe encontrarse antes de que la funcionalidad que expresa el requerimiento se lleve a cabo.</i>
Post condiciones	<i>Cosas que deben ocurrir después</i>
Contactos	<i>Usuario / contacto</i>
Fecha [Tipo]	<i>Tipo de Requerimiento No Funcional. Ej.: Restricción Tecnológica</i>

Requisitos Funcionales

A continuación, se presenta el análisis de requisitos necesarios para el desarrollo de la aplicación web móvil.

Tabla 4. Registro usuario ciudadano, Fuente Autor

RF 001. Registro de Usuario (ciudadano)	
Función	Registro de usuarios (ciudadano)
Descripción	Permitir al usuario registrarse en la aplicación como nuevo usuario del sistema.
Entradas	Datos personales obligatorios: nombres, apellidos, tipo de identificación, número de identificación, correo electrónico, teléfono, lugar de expedición (CC), departamento.
Fuentes	Teclado, archivo
Salidas	Confirmación de usuario registrado, mostrando los datos ingresados.
Proceso	El sistema o la aplicación despliegan el formulario para registro de nuevos usuarios, en el cual se deben ingresar los datos personales – descritos en la entrada. Se validan las entradas y se registra el nuevo usuario.
Restricciones	Si el usuario ya existe, el sistema debe notificar al usuario mostrando los datos actualmente registrados. Si los datos personales no son válidos, entonces el sistema mostrará un mensaje pidiendo que se vuelva a ingresar la información de registro. Por defecto cuando se crea cada usuario,
Precondiciones	
Post condiciones	El usuario queda registrado y puede ingresar posteriormente al sistema o a la aplicación.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 5. Autenticar Usuario, Fuente Autor

RF 002. Autenticar Usuario	
Función	Ingresar usuarios al sistema
Descripción	Permitir a los usuarios ingresar en el sistema utilizando login su contraseña, y recibiendo luego de una autenticación positiva de la APP.

Entradas	User, Password
Fuentes	Teclado
Salidas	Interfaz de usuario acorde con el perfil de los datos de autenticación
. Proceso	Los usuarios ingresan al sistema su login y contraseña, el sistema validará si el usuario existe y su contraseña es correcta. Si es así, el sistema entregará al usuario autenticado la interfaz de usuario correspondiente a su perfil. La contraseña se debe hacer bajo MDS.
Restricciones	Si el login y contraseña no son correctos entonces el sistema muestra un mensaje informando que no son inválidos y que los vuelva a digitar.
Precondiciones	Tener una cuenta en el sistema
Post condiciones	El usuario ingresa al sistema.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 6. Modificar Usuario, Fuente Autor

RF 003. Modificar Usuario	
Función	Permite la actualización de los datos personales de un usuario.
Descripción	Permite a cada usuario modificar sus datos personales modificables, como su contraseña, teléfono, correo electrónico.
Entradas	Datos para modificar, correo electrónico, nombre, teléfono
Fuentes	Teclado
Salidas	Mensaje de confirmación de actualización de datos y visualización de los cambios realizados.

Proceso	<p>El sistema solicita cuál es el usuario a modificar, luego el usuario (ciudadano) ingresa el correo electrónico, el teléfono, identificación o el nombre del usuario para que el sistema lo busque y despliegue los datos del usuario a modificar, el ciudadano realiza los cambios necesarios y envía la petición al sistema para la actualización de la información.</p> <p>El sistema valida los datos a modificar y realiza la actualización de los cambios efectuados.</p>
Restricciones	<p>Datos de identificación -como el login- ante el sistema no deben ser Modificables, así como el nombre. Los datos que son modificables son Teléfono, correo electrónico contraseña. Si el dato o los datos modificados no son válidos, entonces el sistema mostrará un mensaje pidiendo que se vuelva a ingresar la información de modificación.</p>
Precondiciones	El usuario debe haber sido creado en el sistema
Post condiciones	Datos actualizados almacenados en el sistema
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 7. Diligenciamiento de denuncia, Fuente autor.

RF 004. Diligenciar formato de denuncia	
Función	Gestionar formato
Descripción	Permitir al usuario (ciudadano) diligenciar el cuestionario para la identificación de la denuncia, así agregarla al sistema.
Entradas	Cuestionario de denuncia
Fuentes	Teclado

Salidas	Denuncia creada e impuesta por el usuario (ciudadano).
Proceso	El usuario (ciudadano) responde el cuestionario, en el cual evidenciará todas las posibles infracciones que esté presentando el lugar de la incidencia.
Restricciones	Si el usuario no responde el cuestionario el sistema no conseguirá generar la debida denuncia, le notificará que debe responder todas las preguntas para así obtener la denuncia.
Precondiciones	El usuario debe responder todas las preguntas del formato denuncia en el sistema.
Post condiciones	El usuario registró la denuncia.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 8. Gestión de la denuncia, Fuente Autor

RF 005. Gestión de la denuncia	
Función	Diligenciar la denuncia
Descripción	Permite al usuario decidir el manejo de la denuncia
Entradas	Denuncia impuesta.
Fuentes	Teclado,
Salidas	Confirmación de la decisión tomada por el usuario
Proceso	Los ciudadanos deberán confirmar la incidencia a reportar y tomar la opción: Llevar la denuncia a la inspección de policía de la localidad o Enviar a consultorio jurídico.

Restricciones	Si el usuario ya gestionó la incidencia por sus propios medios, el sistema debe notificar al consultorio jurídico que esa denuncia será gestionada por el usuario (ciudadano).
Precondiciones	El ciudadano debe escoger la opción para así llevar acabo la denuncia
Postcondiciones	El ciudadano registra con éxito la opción y es enviada la denuncia a su correo electrónico.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 9. Registro de Imagen, Fuente autor

RF 006. Registrar Imagen	
Función	Agregar foto a la denuncia
Descripción	Permitir al usuario (ciudadano) agregar imágenes del sitio de la incidencia.
Entradas	Imágenes
Fuentes	Archivo jpg, png
Salidas	Imagen agregada.
Proceso	Permitirá el sistema o la aplicación agregar imágenes como soporte para la denuncia impuesta por el ciudadano
Restricciones	Si el usuario no adjunta la imagen el sistema debe notificarle que es necesario adjuntar la imagen, para seguir con el procedimiento de la denuncia
Precondiciones	El usuario debe adjuntar la imagen (foto) en el sistema.
Post condiciones	El usuario almacenar la imagen (foto) en el sistema.
Contactos	

Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 10. Registrar Geolocalización, Fuente autor

RF 007. Registrar Geolocalización	
Función	Ingresar Geolocalización
Descripción	Permitir al usuario (ciudadano) agregar el punto geolocalizado del sitio de la incidencia.
Entradas	Ubicación en el mapa
Fuentes	API GOOGLE
Salidas	Dirección de la incidencia.
Proceso	Permitirá a la aplicación agregar la ubicación en el mapa como soporte para la denuncia impuesta por el ciudadano
Restricciones	Si el usuario no adjunta la ubicación de la incidencia el sistema debe notificarle que es necesario adjuntarla, para seguir con el procedimiento de la denuncia.
Precondiciones	El usuario debe adjuntar la ubicación de la incidencia en el sistema.
Post condiciones	El usuario almacenar ubicación en el sistema.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 11. Consulta de incidencias, Fuente autor

RF 008. Consultar incidencias	
Función	Consultas de denuncias

Descripción	Permitir al usuario Administrador ver las incidencias reportadas por el ciudadano y a quien se le fue asignada, por el equipo de trabajo (docentes, estudiantes)
Entradas	Denuncias con su radicado, notas agregadas
Fuentes	Teclado
Salidas	Confirmación de las denuncias asignadas a los pertinentes grupos de trabajo y radicados
Proceso	El administrador tendrá la capacidad de consultar las denuncias asignadas a los grupos de trabajo del consultorio jurídico, y radicados
Restricciones	El sistema debe mostrar con éxito la búsqueda de la denuncia
Precondiciones	El administrador debe consultar la denuncia por el número de radicado
Post condiciones	El sistema muestra exitoso la búsqueda de la denuncia
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 12. Asignar Incidencias, Fuente autor.

RF 009. Asignar incidencias	
Función	Asignaciones de denuncias
Descripción	Permitir al usuario Administrador las incidencias reportadas por el ciudadano asignarlas al equipo de trabajo (docentes, estudiantes)
Entradas	Denuncias con su radicado
Fuentes	Teclado
Salidas	Confirmación de las denuncias asignadas a los pertinentes grupos de trabajo

Proceso	El administrador tendrá la capacidad de gestionar las denuncias, La asignación a los grupos de trabajo del consultorio jurídico, para así comenzar el debido proceso.
Restricciones	El sistema debe mostrar con éxito la asignación hecha por el administrador
Precondiciones	La aplicación debe tener la denuncia impuesta
Post condiciones	La aplicación asigna la denuncia al usuario (docente) grupo de trabajo
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 13, Lista de Incidencias, Fuente Autor

RF 010. Listar incidencias	
Función	Lista de denuncias
Descripción	Permitir al usuario Administrador ver el histórico de denuncias reportadas por los ciudadanos
Entradas	Denuncias impuestas
Fuentes	Teclado
Salidas	Confirmación del listado de denuncias registradas e ingresadas en el sistema
Proceso	El administrador tendrá la capacidad de ver detalladamente un histórico total de denuncias impuestas por los ciudadanos y la respectiva gestión.
Restricciones	El sistema debe mostrar el histórico general de las denuncias registradas.
Precondiciones	La aplicación debe tener en su base de datos las denuncias impuestas
Post	La aplicación muestra es listado completo de las denuncias

condiciones	
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 14, Registro de usuario Administrador, Fuente Autor

RF 011. Registrar Usuario Administrador	
Función	Ingresar usuarios
Descripción	Permitir al sistema de la aplicación ingresar el usuario registrándose como nuevo usuario del sistema.
Entradas	Datos personales obligatorios: nombre, correo electrónico, teléfono, código.
Fuentes	Teclado, archivo
Salidas	Confirmación de usuario registrado, mostrando los datos ingresados.
Proceso	El sistema o la aplicación despliegan el formulario para registro de nuevos usuarios, en el cual se deben ingresar los datos personales – descritos en la entrada se validan las entradas y se registra el nuevo usuario.
Restricciones	Si el usuario ya existe, el sistema debe notificar al administrador mostrando los datos actualmente registrados. Si los datos personales no son válidos, entonces el sistema mostrará un mensaje pidiendo que se vuelva a ingresar la información de registro. Por defecto cuando se crea cada usuario, la contraseña será su documento de identidad.
Precondiciones	
Post condiciones	El Administrador queda registrado y puede ingresar posteriormente a la aplicación.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 15. Modificación en cuestionario de denuncia, Fuente Autor

RF 012. Modificar Cuestionario de denuncia	
Función	Modificar cuestionario
Descripción	Permitir al usuario (administrador) hacer modificaciones a las preguntas el cuestionario para la denuncia.
Entradas	Datos para modificar
Fuentes	Teclado
Salidas	Mensaje de confirmación de actualización de datos y visualización de los cambios realizados.
Proceso	<p>El sistema solicita cuál es la pregunta para modificar, luego el administrador ingresa el documento que va a hacer modificado, para que el sistema lo busque y despliegue la información a modificar, el administrador realiza los cambios necesarios y envía la petición al sistema para la actualización de la información.</p> <p>El sistema valida los datos a modificar y realiza la actualización de los cambios efectuados.</p>
Restricciones	
Precondiciones	La pregunta debe haber estado creada en el cuestionario de denuncias
Post condiciones	Datos actualizados en el cuestionario.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 16, Autenticación de usuario Docente, Fuente Autor

RF 013. Autenticación de Usuario (Docente)

Función	Registro de usuarios (Docente)
Descripción	Permitir al usuario registrarse en la aplicación como nuevo usuario del sistema.
Entradas	Datos personales obligatorios: nombre, identificación, correo electrónico, teléfono.
Fuentes	Teclado, archivo
Salidas	Confirmación de usuario registrado, mostrando los datos ingresados.
Proceso	El sistema o la aplicación despliegan el formulario para registro de nuevos usuarios, en el cual se deben ingresar los datos personales descritos en la entrada. Se validan las entradas y se registra el nuevo usuario.
Restricciones	Si el usuario ya existe, el sistema debe notificar al usuario mostrando los datos actualmente registrados. Si los datos personales no son válidos, entonces el sistema mostrará un mensaje pidiendo que se vuelva a ingresar la información de registro. Por defecto cuando se crea cada usuario,
Precondiciones	
Post condiciones	El usuario queda registrado y puede ingresar posteriormente al sistema o a la aplicación.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 17. Consultar asignaciones de incidencias, Fuente autor

RF 014. Consultar Asignaciones de Incidencias	
Función	Consultas de asignación de denuncias
Descripción	Permitir al usuario (Docente) ver las incidencias asignadas por el Administrador.
Entradas	Denuncias con su radicado, observaciones

Fuentes	Teclado
Salidas	Confirmación de recibido de las denuncias asignadas pertinente a sus grupos de trabajo.
Proceso	El docente tendrá la capacidad de consultar y gestionar las denuncias asignada a su grupo de trabajo del consultorio jurídico.
Restricciones	El sistema debe mostrar el listado de la denuncia asignada a su grupo de trabajo, de lo contrario el sistema le notificará que aún no tiene denuncias asignadas.
Precondiciones	El docente debe consultar las denuncias asignadas
Post condiciones	El sistema muestra exitoso la búsqueda del listado de las denuncias.
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 18. Generar Reportar sobre incidencias, Fuente autor.

F 015. Generar reportes de incidencias	
Función	Gráfica de incidencias
Descripción	Permitir al usuario (administrador) imprimir en pantalla de modo que se reflejen las denuncias recientes.
Entradas	Las denuncias impuestas y registradas en el sistema
Fuentes	Teclado
Salidas	Gráficas de las denuncias recientes
Proceso	El sistema tomará toda la información de las denuncias recientes y hará una gráfica estadísticamente donde obtendrá los reportes semanales de las denuncias realizadas en la aplicación.
Restricciones	

Precondiciones	La aplicación debe tener en su base de datos todas las denuncias impuestas
Post condiciones	
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

Tabla 19. Chat Bot, Fuente Autor

RF 016. Chat Bot	
Función	Permitir la Interacción entre el ciudadano y consultorio jurídico.
Descripción	Permitir al usuario administrador y docente, visualizar y contestar todas las preguntas o inquietudes de los ciudadanos.
Entradas	Las preguntas impuestas y registradas en el sistema.
Fuentes	Teclado
Salidas	Solución a inquietudes
Proceso	El sistema tomará toda la información de las preguntas de los ciudadanos, el administrador se encargará de dar solución a las inquietudes sobre la realización de las denuncias.
Restricciones	
Precondiciones	El ciudadano debe hacer las preguntas e inquietudes a la aplicación
Post condiciones	El administrador responderá las inquietudes de los ciudadanos
Contactos	
Fecha [Tipo]	14-03-2019 Esencial y primario.

3. DISEÑO DEL SISTEMA

En este capítulo se encuentra detallado el diseño de la aplicación web móvil para la realización y gestión de denuncias urbanística. Entre los diseños encontramos casos de uso para cada uno de los autores relevantes, diagrama de secuencia, diagrama de despliegue, diagrama de proceso, diagrama de componentes, diagrama de despliegue, diagrama de distribución; además de los diagramas de clases. También se describen los diseños de cada una de las interfaces de usuario de acuerdo con los requerimientos del sistema. A continuación, se darán a conocer los casos de uso

Caso de uso: Ciudadano

Como se observa en la ilustración 7, encontramos el caso de uso en el cual el ciudadano debe estar registrado en la aplicación o en su defecto registrarse para luego poder hacer uso de cada una de las funcionalidades del módulo como realizar la denuncia. Este caso de uso está relacionado con los requerimientos: RF001, RF002, RF003, RF004, RF005, RF006, RF007, RF008.

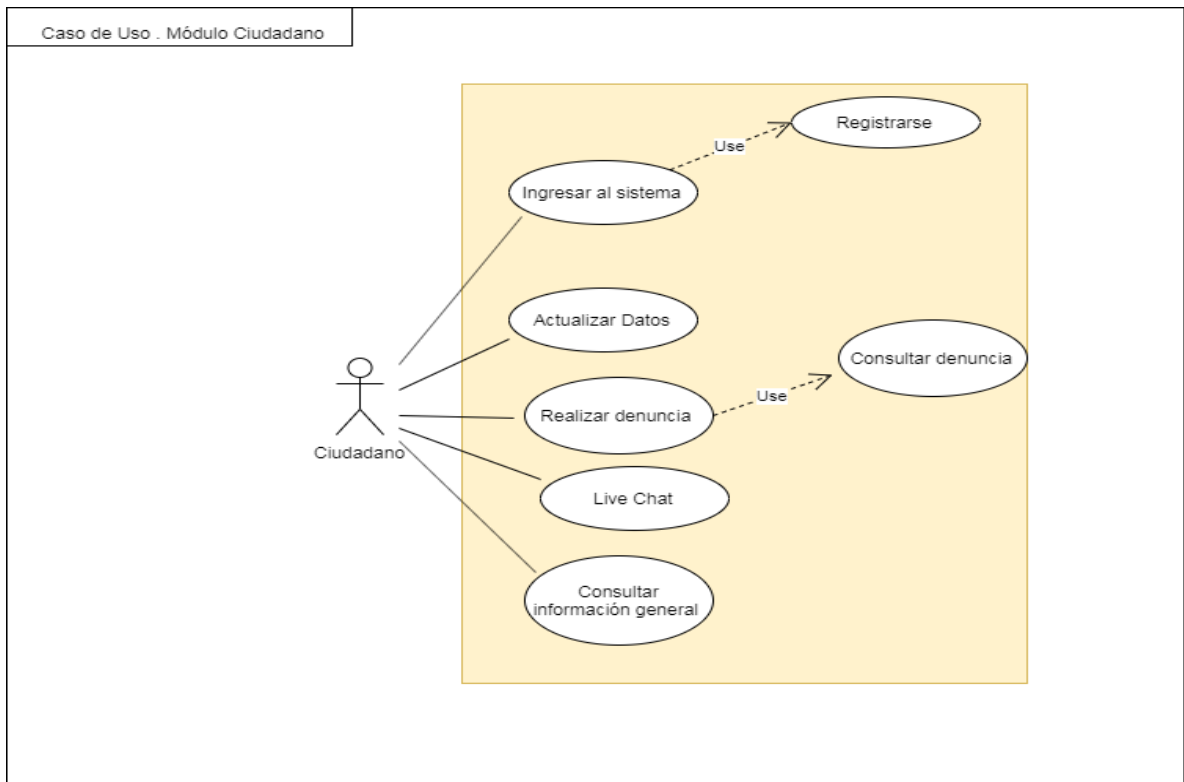


Ilustración 7.Caso de un uso ciudadano, Fuente autor

Caso de uso: Administrador

En este caso de uso vemos en la ilustración número 8, las funcionalidades en módulo de administración del sistema; cuenta desde el inicio de sesión, listado de denuncias generadas hasta la asignación de cada una de las denuncias a los respectivos docentes. Este caso de uso se relaciona con los siguientes requerimientos del sistema: RF009, RF010, RF011, RF012

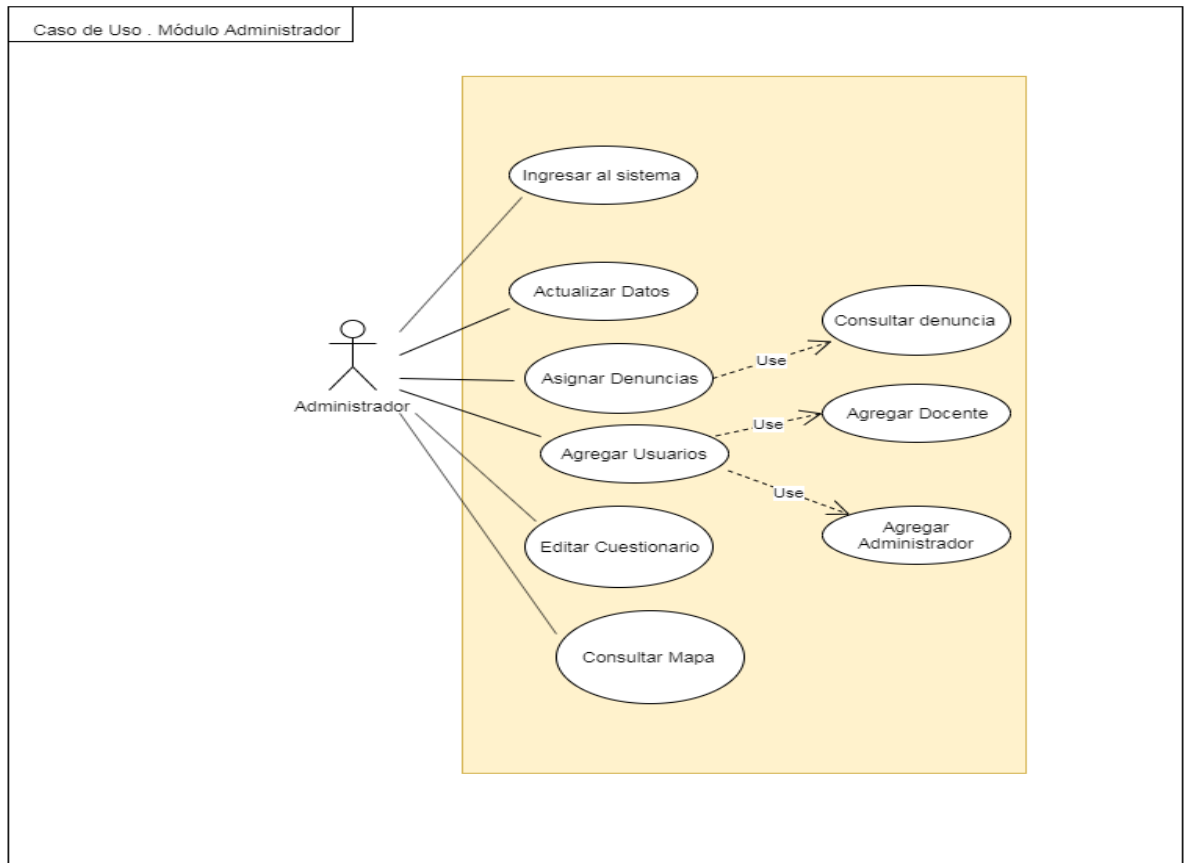


Ilustración 8. Caso de uso Administrador. Fuente autor

Caso de uso: docente

En este caso de uso se detallan las funcionalidades del módulo docente, para la gestión de las denuncias previamente asignadas por un administrador, al igual que la interacción con el ciudadano al agregar las notas de la gestión de las denuncias como lo muestra la ilustración 9. Este caso de uso se relaciona con los siguientes requerimientos del sistema: RF013, RF014

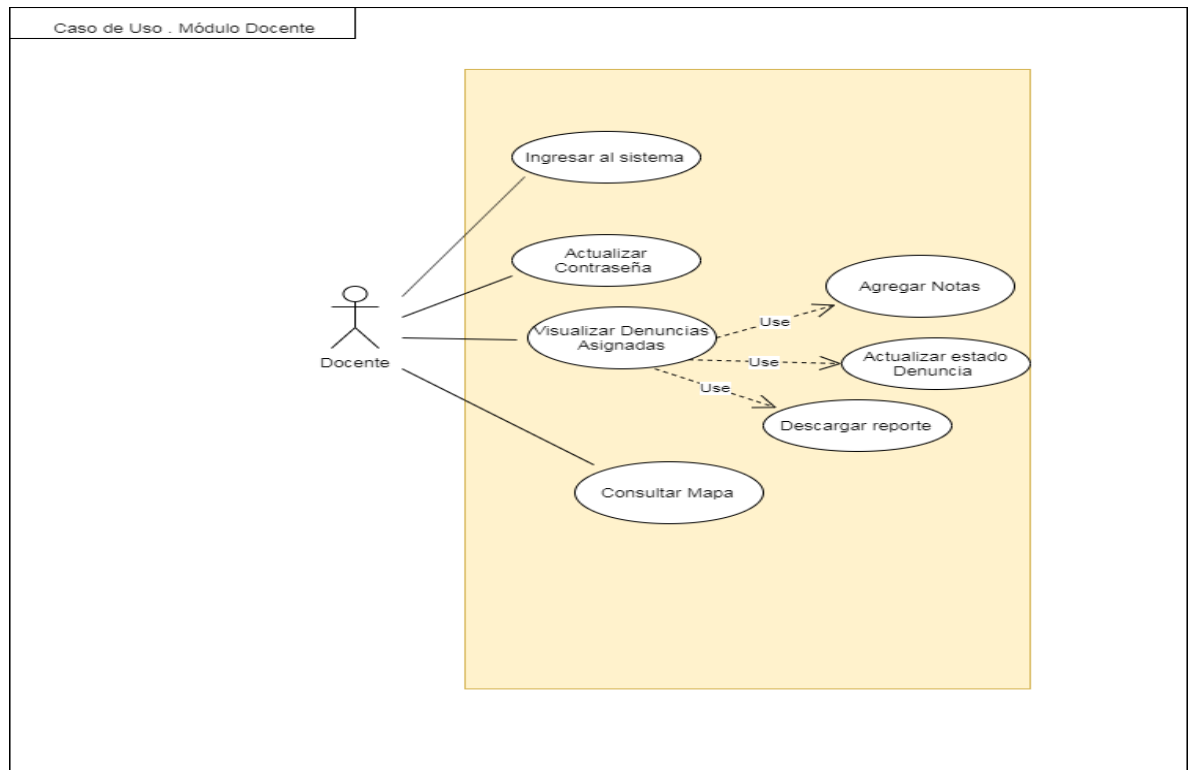


Ilustración 9. Caso de uso Docente, Fuente autor

3.1 Diagrama de secuencia

Diagrama de secuencia: Ciudadano

La ilustración 10 describe el diagrama de secuencia de la interacción de los objetos a través del tiempo, la imagen describe los pasos al realizar la denuncia y la petición que se realiza a la base de datos y la respuesta de la misma, las interfaces de error y confirmación de la denuncia realizada.

Diagrama Secuencial Ciudadano

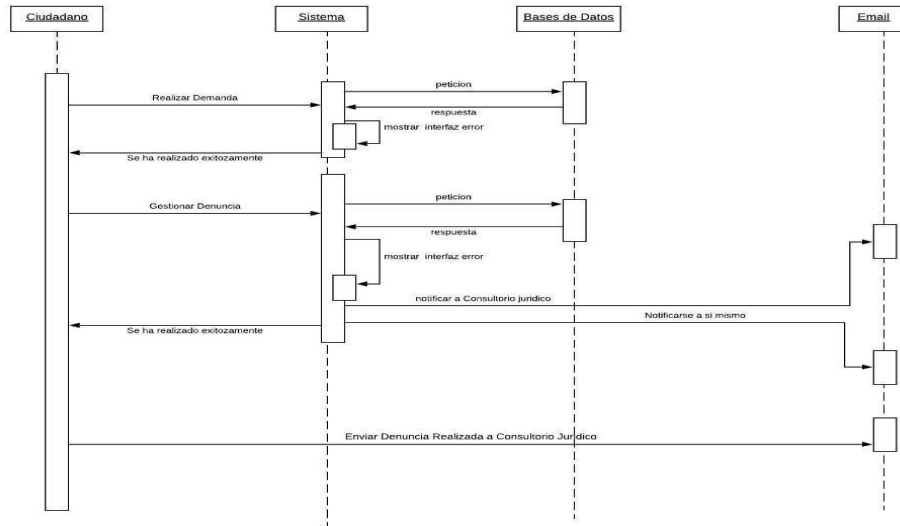


Ilustración 10. Diagrama de secuencia Ciudadano, Fuente autor

Diagrama de secuencia: Administrador

La ilustración 11, describe el diagrama de secuencia de la interacción de los objetos a través del tiempo se ilustra el CRUD al realizar el registro de un usuario y asignación de las denuncias.

Diagrama Secuencial Admin

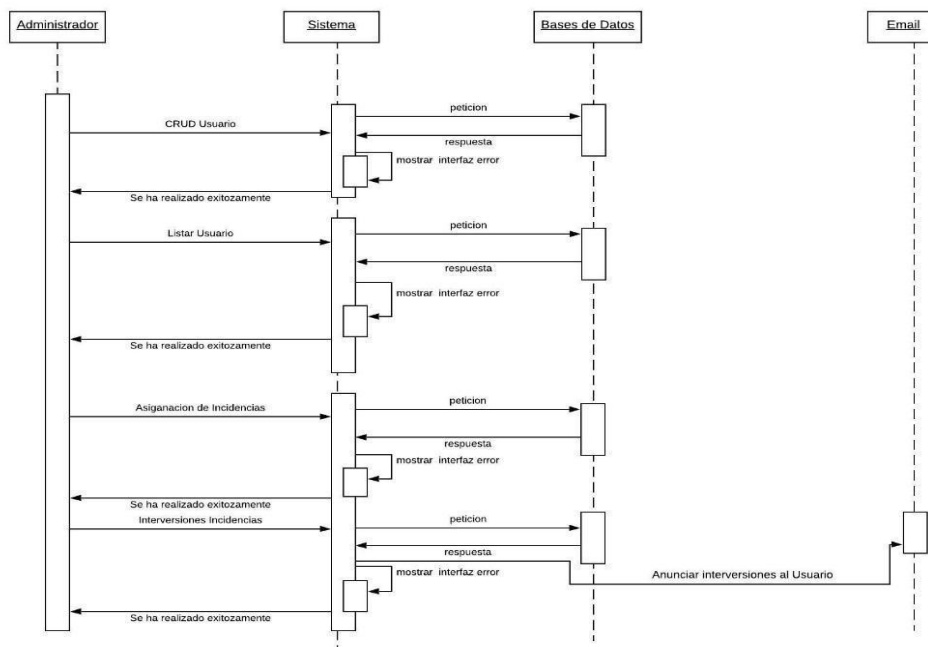


Ilustración 11. Diagrama de secuencia Administrador, Fuente autor

Diagrama de secuencia: Docente

La ilustración 12, describe el diagrama de secuencia de la interacción de los objetos a través del tiempo en el proceso de gestión de la denuncia, la petición a la base de datos y la respuesta de esta.

Diagrama Secuencial Docente

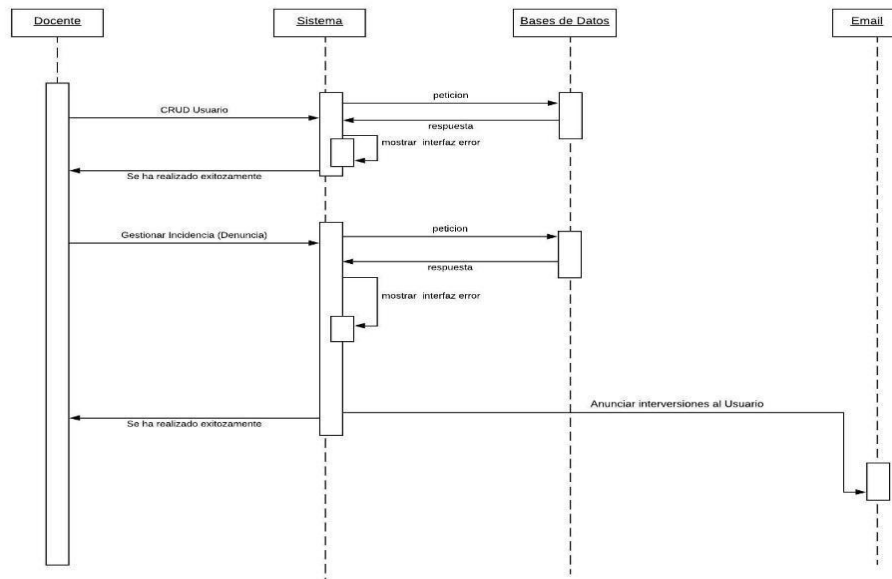


Ilustración 12. Diagrama de secuencia Docente, Fuente autor

3.2 Diagrama de procesos

Diagrama de procesos, módulo administrador

La ilustración número 13, describe la secuencia de los procesos del módulo administrador, teniendo en cuenta el proceso de “logeo”, listar las denuncias generadas, el proceso de asignación, creación de usuario docente y administrador, actualización de datos de usuario, consulta de mapa de denuncias y la administración del formulario.

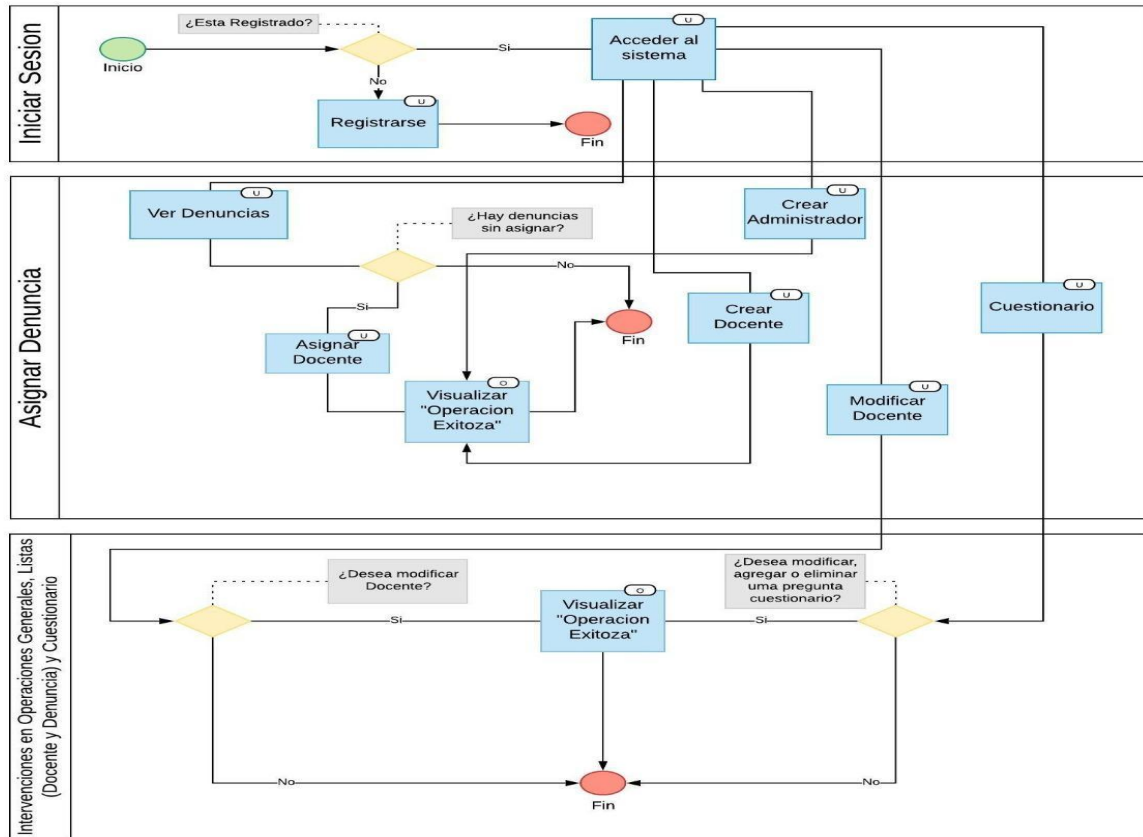


Diagrama de procesos, módulo ciudadano

El diagrama de procesos del módulo ciudadano cuenta con los procesos de Registro, inicio de sesión una vez culminado el registro, Actualización de datos de usuario, creación de denuncia, diligenciamiento del formato de la denuncia más la carga de imágenes relacionadas al caso, consulta de denuncias generadas, consultar la información general del sistema y la interacción a través del chat con el personal de consultorio jurídico.

Ciudadano

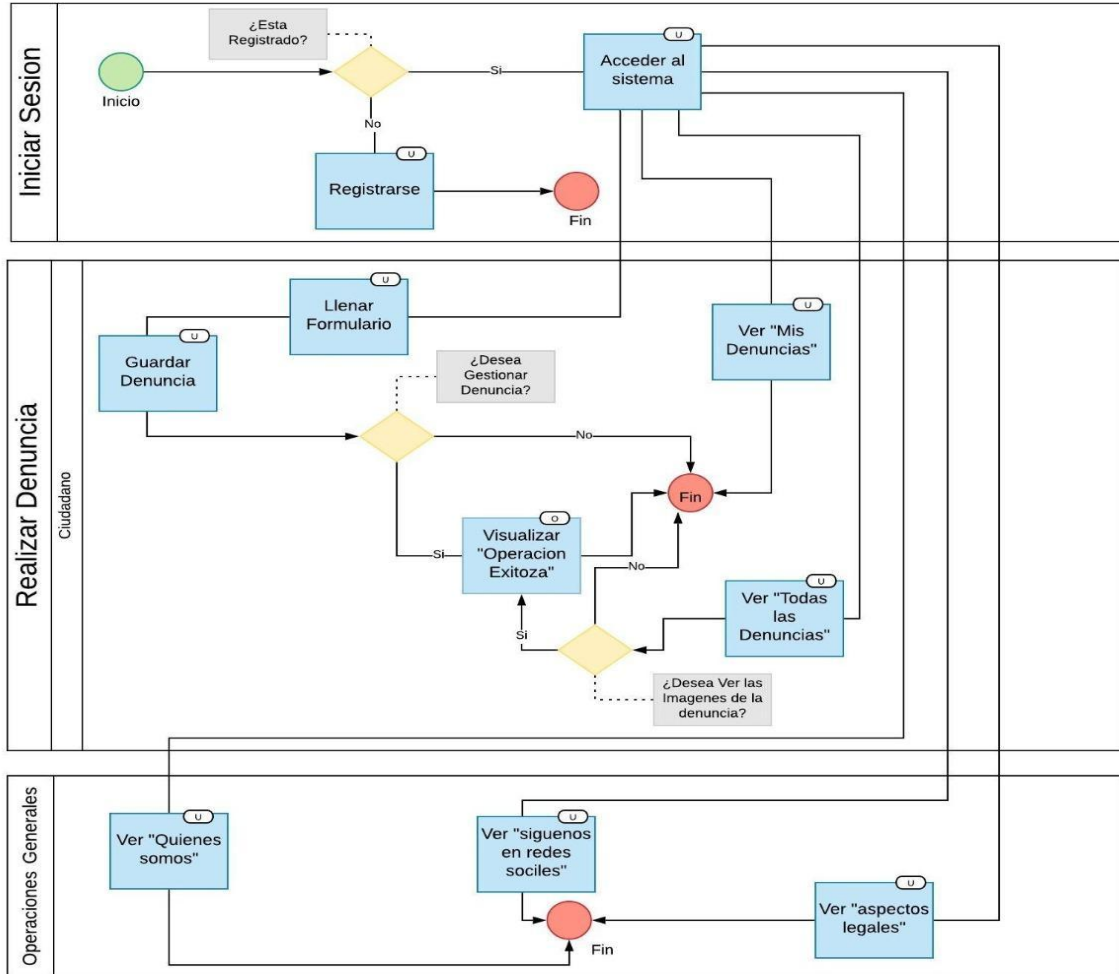


Ilustración 14. Diagrama de secuencia Ciudadano, Fuente autor

Diagrama de procesos, módulo docente

En el diagrama de procesos del módulo Docente se describen los procesos de autenticación de usuario, consulta de denuncias asignadas por el administrador, gestión de las denuncias agregando notas a cada caso según la evolución de estas, actualizar

el estado de la denuncia, consultar mapa de caso y el cambio de contraseña (ver ilustración 15).

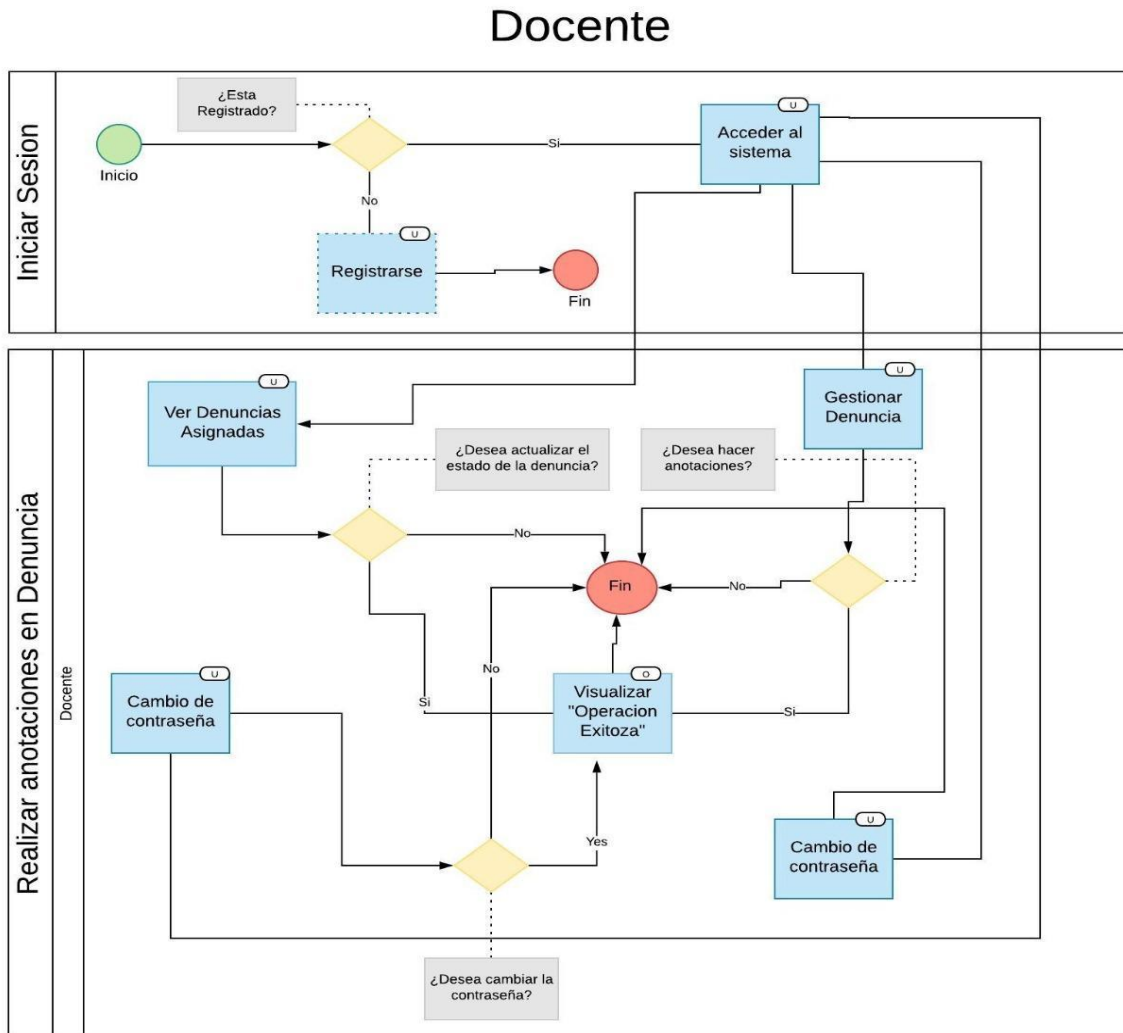


Ilustración 15. Diagrama de secuencia Docente, Fuente autor

3.3 Diagrama de componentes

El siguiente diagrama describe los componentes de la plataforma, esta cuenta con la capacidad de adaptarse a cualquier dispositivo ya sea Tablet o Teléfono móvil, cada componente interactúa entre sí, cada petición es solicitada a la base de datos y los datos son mostrados en pantalla. De la misma manera ocurre con la interfaz web (ver Ilustración 16)

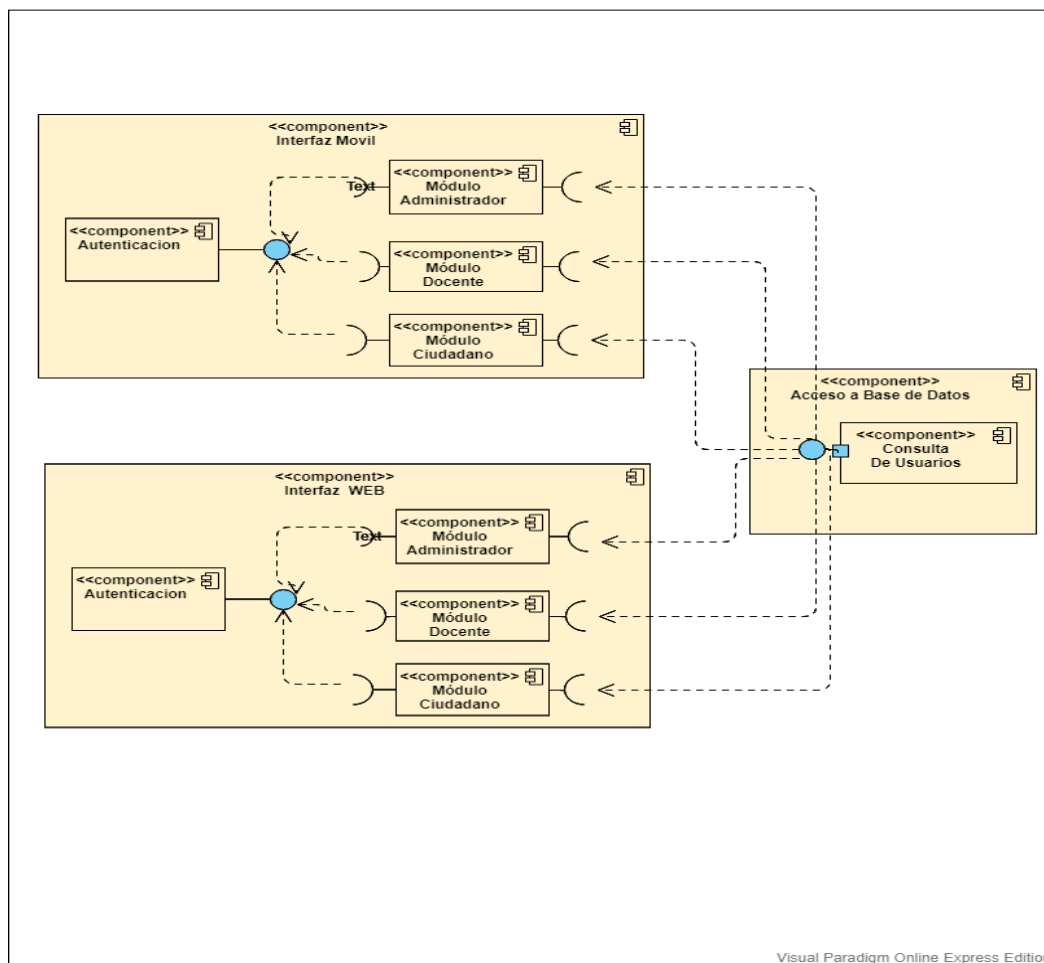


Ilustración 16. Diagrama de componentes, Fuente autor

3.4 Diagrama de despliegue

El diagrama de despliegue ilustra las relaciones físicas de los distintos nodos que componen gestor urbano y el reparto de los componentes sobre dichos nodos. Se utilizan los protocolos TCP/IP y HTTP/HTTPS. El sitio se encuentra alojado en un servidor con base de datos MYSQL (Ver Ilustración 17).

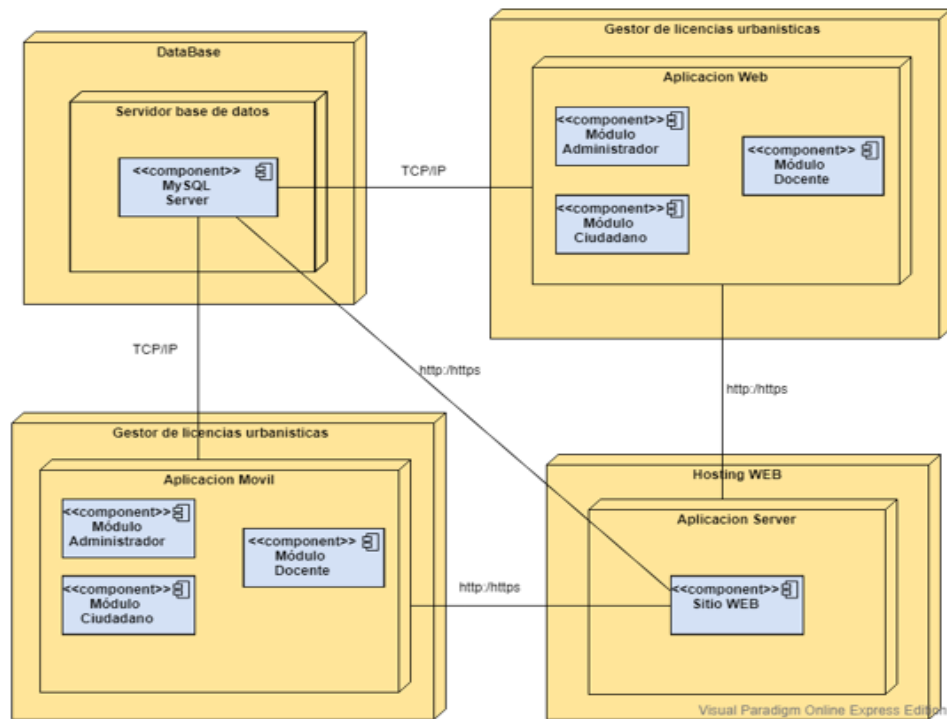


Ilustración 17. Diagrama de despliegue, Fuente autor

3.5 Diagrama de distribución

El siguiente diagrama de distribución se representa la estructura de cómo se encuentra nuestro software representado por los nodos, como referencia se toman los dispositivos móviles, PC's y el servidor de base de datos (Ver Ilustración 18).

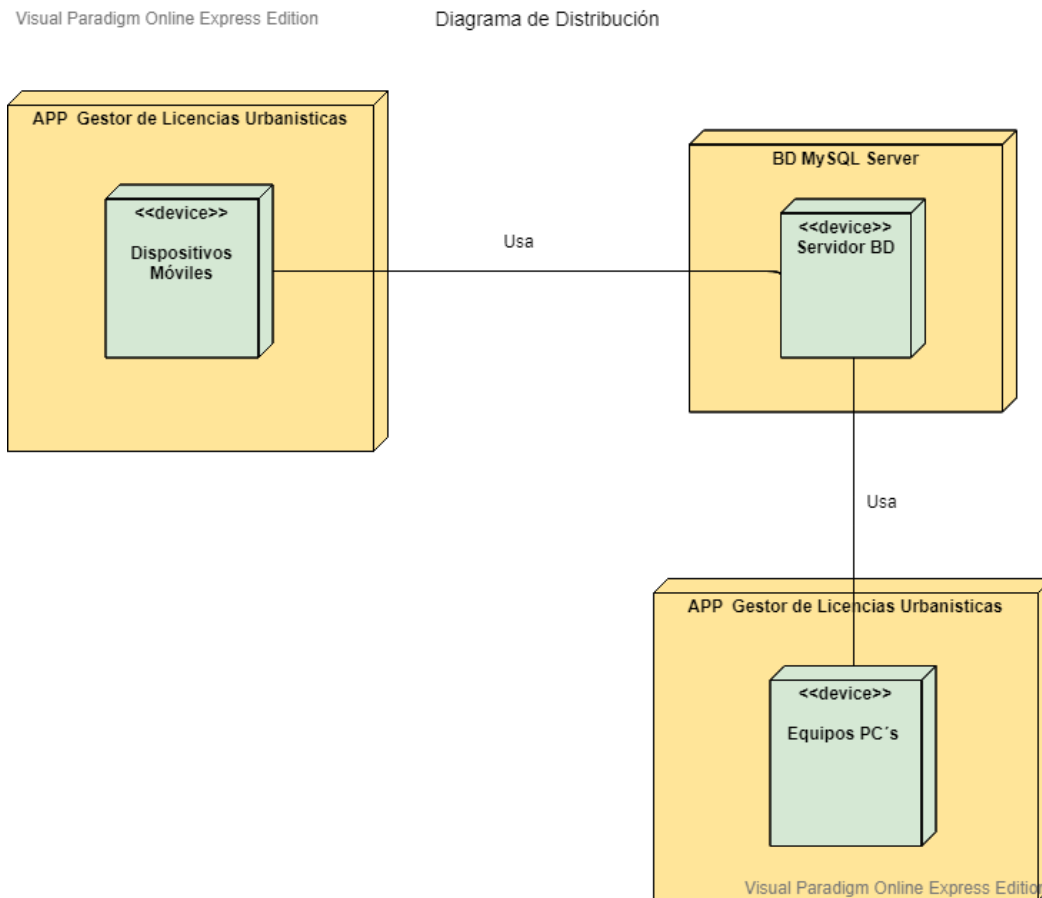


Ilustración 18. Diagrama de distribución, Fuente autor

3.6 Diagrama Entidad Relación

El modelo relacional nos permite analizar la estructura de la base de datos y la relación que hay entre cada uno de los datos al igual que permitimos analizar cómo interactúan dichos datos entre sí. El modelo representado en la ilustración número 19, fue diseñado con la herramienta Lucid chart.

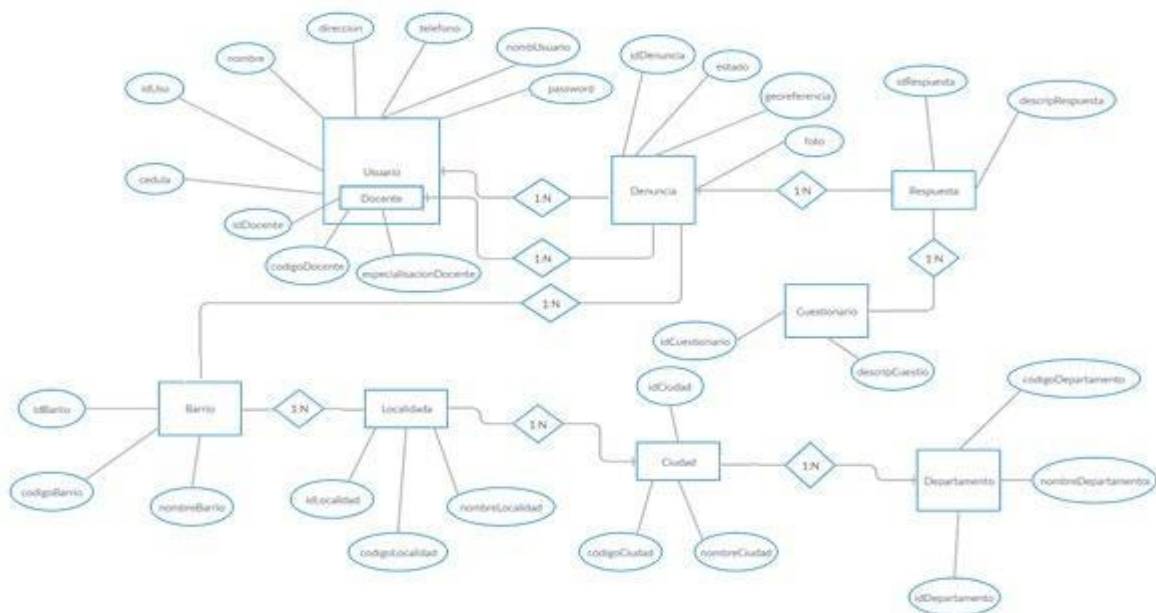


Ilustración 19. Modelo Entidad Relación, Fuente autor

Modelo relacional

El modelo relacional al igual que el de entidad relación, también nos permite analizar la información de una base de datos de una forma visualmente agradable y sencilla, como también la información que esta contiene, y nos da la ventaja de analizar la información en equipo. En la ilustración número 20 podemos ver cada una de las relaciones y detalles de las tablas con sus entidades realizado con la ayuda de la herramienta MySQL Worbech 8.0.

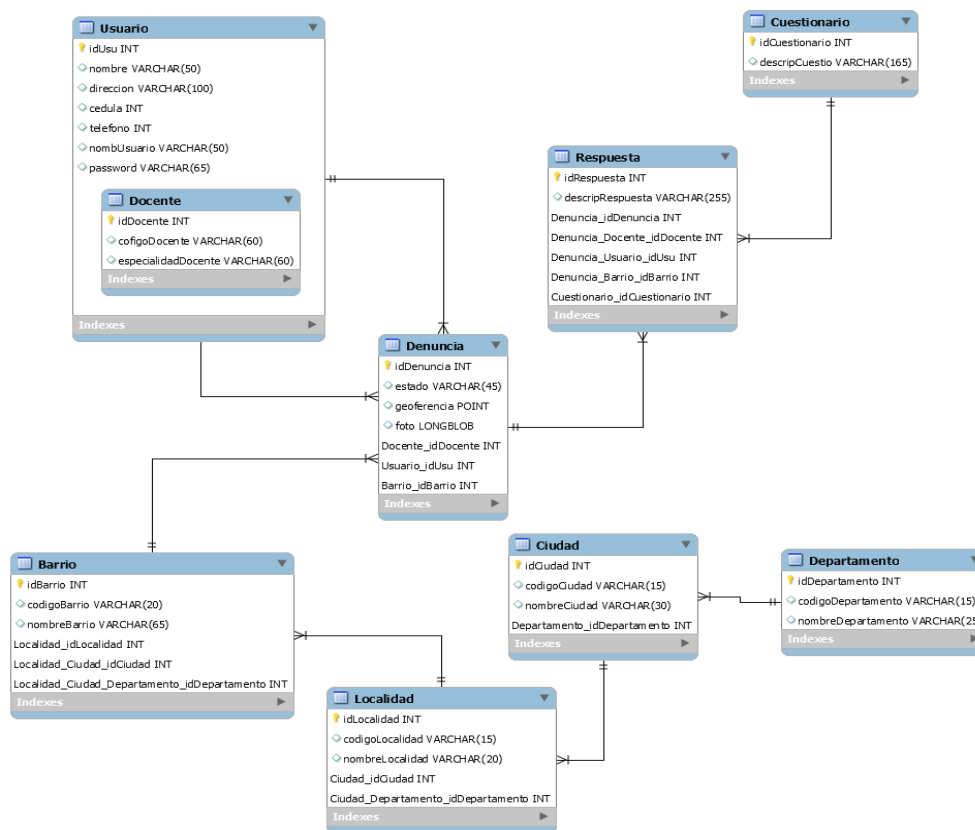


Ilustración 20. Modelo Relacional, fuente autor

Diagrama de Clases

El siguiente diagrama UML es un tipo de diagrama de estructura estática que describe la estructura del sistema mostrando las clases, sus atributos, operaciones (o métodos), y las relaciones entre los objetos. (ver ilustración 21).

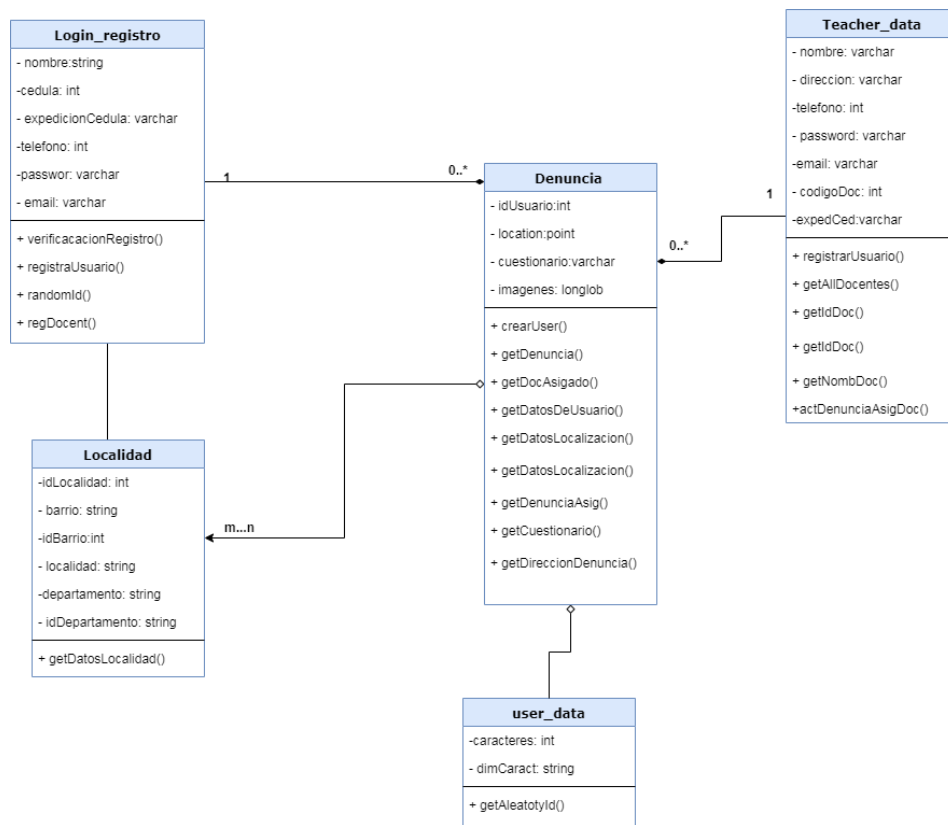


Ilustración 21. Diagrama Relacional, fuente Autor

3.7 Diseño de interfaces de usuario

Inicio de sesión

El inicio de sesión como lo ilustra la Ilustración 22, encontramos las siguientes características: Inicialmente el logo de la aplicación, luego el tipo de cuenta y la opción de seleccionar ya sea como Administrador, docente o en nuestro caso Ciudadano. Luego contamos con el campo para el correo electrónico y posteriormente el campo para la contraseña.

Si el usuario no se encuentra registrado, está opción de registrarse o en el caso de recordar la contraseña.



Ilustración 22. Inicio de sesión, fuente autor

Tabla 20. Inicio de sesión, formato REQ_VIEW_V1

Inicio de sesión			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	Logo de la app	Tooltip	No aplica
Descripción*	Logo de del prototipo		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Tipo de cuenta		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo box	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Tipo de usuario seleccionar para el inicio de sesión		
Rango de valores permitidos	Lista predeterminada		
Observaciones*	No aplica		
Campo*:	Correo		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Dirección de correo electrónico del usuario.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Contraseña		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Contraseña con la que el usuario accederá al sistema.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Regístrese		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Cuadro de texto	Tipo de dato	texto
Valor por defecto	No aplica	Tooltip	No aplica

Descripción*	Esta opción tiene la funcionalidad de permitir al usuario registrarse en la plataforma		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Olvido su contraseña		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Link	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Link para restablecer contraseña del usuario y posteriormente ser enviada al correo del mismo.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Ingresar		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Botón	Tipo de dato	No aplica
Valor por defecto	Ingresar	Tooltip	No aplica
Descripción*	Acción por medio el usuario ingresa al sistema		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Registro de usuario

En el registro de usuarios el sistema permite al ciudadano ingresar sus datos en la aplicación para que estos sean almacenados en base de datos. Los datos solicitados son relevantes para las posteriores denuncias, ya que el documento final (la denuncia) toma los estos datos del usuario y los presenta en el PDF.

Los campos solicitados y obligatorio son: Nombre completo, dirección, cedula, lugar de expedición, Teléfono, Contraseña y confirmación de contraseña para efectos de calidad. Tal y como lo muestra la Ilustración 23.

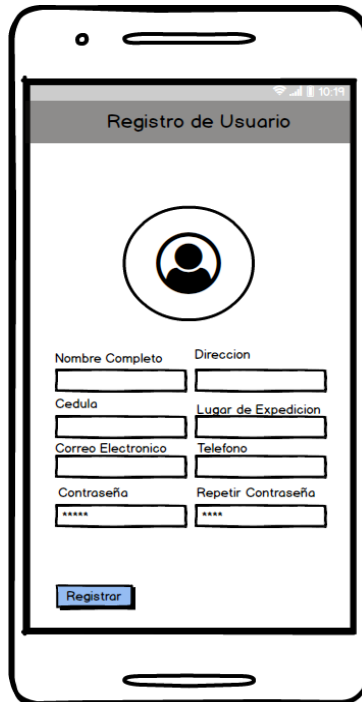


Ilustración 23.Registro de usuario, fuente autor

Tabla 21. Registro de usuario, formato REQ_VIEW_V1

<i>Registro de usuario</i>			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Logo de la empresa		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Nombre completo		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Nombres completos del usuario con el que se establecerán las denuncias.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Campo*:	Dirección		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Dirección de residencia del Usuario		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Cedula		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	int
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Número de identificación del usuario		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Lugar de expedición		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Texto	Tipo de dato	texto
Valor por defecto	No aplica	Tooltip	Si
Descripción*	Lugar de expedición del documento del usuario		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Correo Electrónico		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Campo de texto	Tipo de dato	texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Correo electrónico del usuario con el cual posteriormente iniciará sesión en la aplicación		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Teléfono		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Dirección de residencia del Usuario		

Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Contraseña		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Contraseña de ingreso a la aplicación		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Repetir contraseña		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	Si
Descripción*	Acción para validar la contraseña ingresada previamente		
Rango de valores permitidos	No aplica		
Observaciones*	La contraseña ingresada debe coincidir con la primera		

Recuperar contraseña

En esta vista el usuario ciudadano puede realizar la recuperación de la contraseña en caso de que este la haya olvidado. Para ello deberá contar con un registro previo, ya que el sistema validará el correo electrónico registrado previamente y luego enviará un link de recuperación a la cuenta del usuario con la nueva contraseña. Es como se muestra ilustrado en la Figura 24.

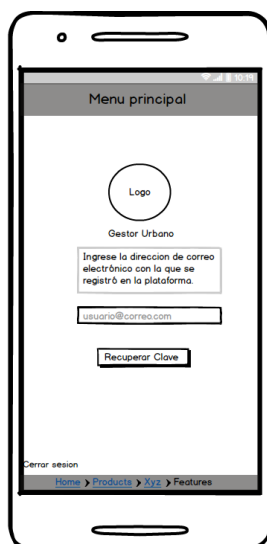


Ilustración 24. Recuperar Contraseña, fuente autor

Tabla 22. Recuperar Contraseña, formato REQ_VIEW_V1

Recuperar Contraseña			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Logo de la empresa		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Ingreso de correo		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Campo de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	Si
Descripción*	Campo para ingresar la dirección de correo electrónico para recuperación de contraseña		
Rango de valores permitidos	Correo electrónico		
Observaciones*	La contraseña será enviada al correo que fue registrado		
Campo*:	Recuperar Clave		
Longitud	No aplica	Obligatorio*	si
Tipo control*	Botón	Tipo de dato	No aplica

Valor por defecto	Recuperar Clave	Tooltip	No aplica
Descripción*	Acción por la cual es enviado correo electrónico con link de recuperación de contraseña		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Menú principal

El menú principal fue diseñado con el fin de brindarle una experiencia al usuario visualmente agradable, de tal manera que no se vea recargado de ilustraciones y botones o accesos. De esta forma será mucho más intuitivo. Por un lado; en la parte izquierda tenemos el menú con cada uno de los iconos que generan las diferentes acciones como: Crear denuncia, Mis denuncias, preguntas frecuentes quienes somos y otros datos relevantes como podemos ver en la Ilustración 25.

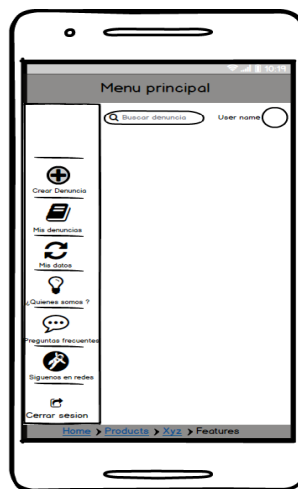


Ilustración 25. Menú principal, fuente autor

Tabla 23. Menú principal, formato REQ_VIEW_V1

Menú principal	
Campo*:	Iconos

Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	Logos	Tooltip	No aplica
Descripción*	El menú principal muestra los botones a la izquierda con la descripción de los mismos		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Buscar denuncias		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Campo de búsqueda	Tipo de dato	Texto
Valor por defecto	Buscar denuncia	Tooltip	No aplica
Descripción*	Campo para realizar la búsqueda de denuncias realizadas por el usuario.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Nombre de usuario		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	Nombre del usuario	Tooltip	No aplica
Descripción*	Nombre del usuario registrado en el sistema		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Crear incidencia

En esta vista observamos el proceso de creación de la denuncia. Esta cuenta con una serie de preguntas previamente configuradas en el sistema, relacionadas a ciertos aspectos relevantes en el proceso de obras en construcción. Algunas de las preguntas como cantidad de pisos de la obra, si cuenta con parqueadero o si esta se construye sobre el espacio público. El usuario debe marcar sí o no en cada una de ellas; todas las preguntas son de carácter obligatorio lo que indica que deben estar marcadas ya sea con si o no.

Además, este solicita al usuario ciudadano si desea gestionar la denuncia o bien, dejar que consultorio jurídico de la universidad del Sinú sea quien se encargue de todo el proceso tal y como lo ilustra la figura 26.

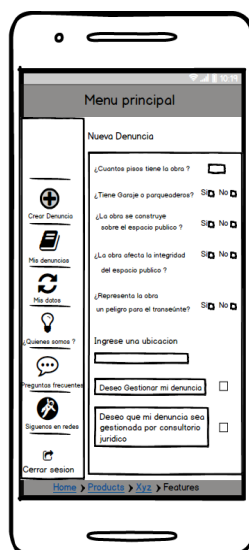


Ilustración 26. Nueva incidencia, fuente autor

Tabla 24. Nueva incidencia, formato REQ_VIEW_V1

Nueva incidencia			
Campo*:	Nueva denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Formulario	Tipo de dato	Bolean
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Formulario de preguntas con dos respuestas (Si /No) en el cual el usuario deberá contestar según sea el caso a registrar		
Rango de valores permitidos	Pregunta formulada		
Observaciones*	No aplica		
Campo*:	Ingresar ubicación		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica

Descripción*	Ubicación, barrio o establecimiento donde se genera la denuncia		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Deseo gestionar mi denuncia.		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Check box	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción por la cual el ciudadano decidirá si el mismo desea realizar el trámite de la denuncia.		
Rango de valores permitidos	No aplica		
Observaciones*	El ciudadano recibirá un correo con la denuncia		
Campo*:	Gestionar denuncia CJ (Consultorio Jurídico)		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Check box	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción por la cual el ciudadano decide si desea que el consultorio jurídico sea quien gestione su caso.		
Rango de valores permitidos	No aplica		
Observaciones*	El caso se enlista para ser gestionado por el CJ		

Además de lo anterior descrito el usuario ciudadano puede adjuntar un máximo de tres imágenes como evidencia del caso, las cuales podrán ser consultadas como anexos a la denuncia, al igual que la geolocalización del lugar donde se origina la denuncia tal y como lo muestra la ilustración 27.

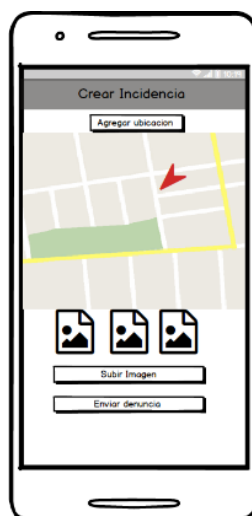


Ilustración 27. Cargar imágenes, fuente autor

Tabla 25. Cargar imágenes, formato REQ_VIEW_V1

Cargar imágenes			
Campo*:	Agregar ubicación		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Selección	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Acción por la cual el sistema tomará la geolocalización del punto donde se genera la denuncia.		
Rango de valores permitidos	No aplica		
Observaciones*	El sistema dejará registrado el punto donde se genera la denuncia		
Campo*:	Subir imagen		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	Icono imagen	Tooltip	No aplica
Descripción*	Ubicación, barrio o establecimiento donde se genera la denuncia		

Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	No aplica		
Campo*:	Enviar denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Botón	Tipo de dato	Texto
Valor por defecto	Enviar denuncia	Tooltip	No aplica
Descripción*	Acción por la cual es enviada la denuncia y son guardados los datos en la BD		
Rango de valores permitidos	No aplica		
Observaciones*	Se muestra mensaje resumiendo el envío incluyendo el consecutivo de la denuncia.		

Mis denuncias

En esta vista (Ilustración 28) el usuario ciudadano puede encontrar el listado de las incidencias generadas, además de consultarlas, este puede descargarlas y realizar el seguimiento de estas, validando las notas agregadas por el personal asignado de la gestión (Ver Ilustración 29).

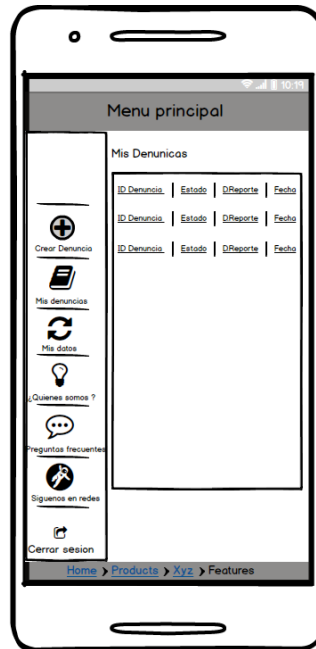


Ilustración 28. Mis denuncias, fuente autor

Tabla 26. Mis denuncias, formato REQ_VIEW_V1

Mis denuncias			
Campo*:	Id denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Consecutivo de la denuncia		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Estado denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Botón	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Muestra el estado de la denuncia por un color en específico		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Campo*:	Descargar reporte		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	PDF	Tipo de dato	Texto
Valor por defecto	Boton	Tooltip	No aplica
Descripción*	Muestra la denuncia en formato PDF con todos los datos		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Atendido por		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo Box	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Se muestra el listado con los docentes disponibles		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Fecha de la denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Date	Tipo de dato	Date
Valor por defecto	Fecha denuncia	Tooltip	No aplica
Descripción*	Se muestra el listado		
Rango de valores permitidos	No aplica		
Observaciones*	El sistema toma la fecha en el momento de la realización de la denuncia.		

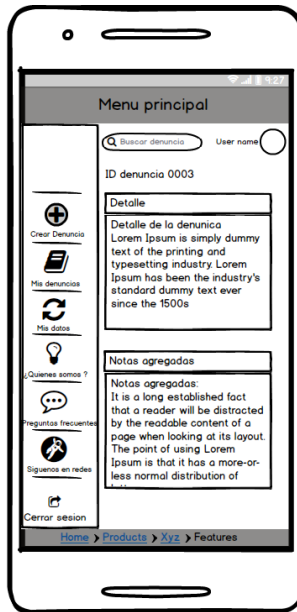


Ilustración 29. Consulta de incidencias, fuente autor

Vista de denuncias

La denuncia final se verá transcrita en formato PDF con todos los datos legales, además de la información suministrada por el usuario ciudadano, al igual que la dirección del lugar donde se genera y los aspectos a intervenir en el sitio, representado en la ilustración 30.

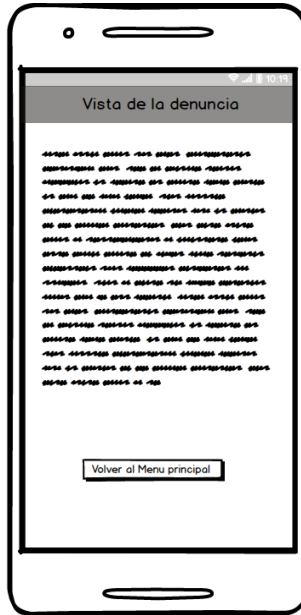


Ilustración 30. Vista de la denuncia, fuente autor

Tabla 27. Vista de la denuncia, formato REQ_VIEW_V1

Vista de la denuncia			
Campo*:	Vista de la denuncia		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	PDF	Tipo de dato	Texto
Valor por defecto	PDF	Tooltip	No aplica
Descripción*	Acción por la cual el usuario puede visualizar y descargar la denuncia realizada en formato PDF		
Rango de valores permitidos	No aplica		
Observaciones*	La denuncia contiene las respuestas al cuestionario realizado y los datos del usuario como Cédula y Lugar de expedición.		

Información general

Esta vista (Ilustración 31) cuenta con información relevante acerca de la Aplicación en general, con datos como: Quienes somos, sistema de participación ciudadana, preguntas frecuentes y las redes salicales asociadas a la misma. De esta forma el usuario ciudadano puede consultar y profundizar acerca de la finalidad de la aplicación, de tal modo que este se sienta incluido dentro de los procesos que competen en el desarrollo urbanístico.

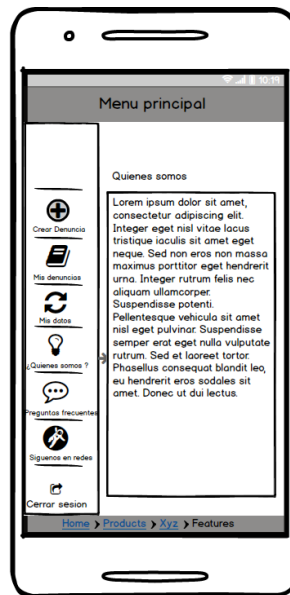


Ilustración 31. Opciones generales, fuente autor

Tabla 28. Opciones generales, formato REQ_VIEW_V1

Opciones generales			
Campo*:	Quiénes Somos		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Campo de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Información relevante sobre la aplicación		
Rango de valores permitidos	No aplica		

Observaciones*	No aplica		
Campo*:	Preguntas Frecuentes		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Campo de texto	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Preguntas relacionadas a la aplicación en las cuales el usuario podrá aclarar ciertas dudas		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Síguenos en redes		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Campo de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Lista de redes sociales asociadas a la aplicación		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Sistema de participación ciudadana		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Campo de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Muestra información relevante acerca del sistema de participación ciudadana.		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Página principal

Esta vista de inicio (Ilustración 32) es la ventana principal de la plataforma en donde el usuario encontrará a disposición cada uno de los botones los cuales generan una acción específica, además de ofrecer una interfaz amigable y de uso intuitivo para una mejor experiencia de usuario.

Encontramos el logo del aplicativo en el lado izquierdo al igual que el rol del usuario. En la parte superior derecha encontramos el nombre del usuario logueado y la disposición de botones alineados a la izquierda.



Ilustración 32. Vista Principal Administrador, fuente autor

Gestionar denuncia

En esta vista (Ilustración 33) observamos el listado de casos generados por los ciudadanos. El administrador puede visualizar y gestionar cada uno de estos. Cuenta con los datos de Id Denuncia el cual es un consecutivo generado automáticamente una vez se realice la denuncia, el estado denuncia el cual muestra una serie de colores dependiendo el estado de esta. (Iniciado, en curso, finalizada, rechazada). Descargar

Reporte donde podemos visualizar y descargar la denuncia en su formato establecido con todos los datos diligenciados por el usuario Ciudadano, Atendido por donde podemos ver la persona encargada del caso o bien; asignar un docente al mismo y por último la fecha en la que fue generada la denuncia.

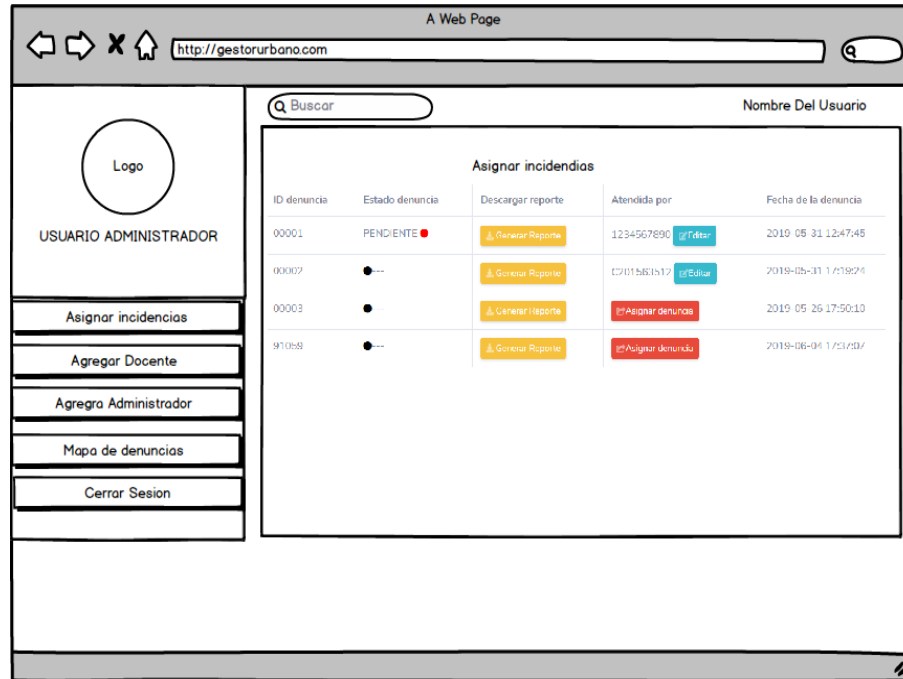


Ilustración 33. Asignar caso a Docente, fuente autor

Tabla 29. Asignar caso a Docente, formato REQ_VIEW_V1

Asignar caso a Docente			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	lg	No aplica
Descripción*	Logo de la institución		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		

Campo*:	Id denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Consecutivo de la denuncia		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Estado denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Botón	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Muestra el estado de la denuncia por un color en específico		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Descargar reporte		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	PDF	Tipo de dato	Texto
Valor por defecto	Boton	Tooltip	No aplica
Descripción*	Muestra la denuncia en formato PDF con todos los datos		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Atendido por		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo Box	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Se muestra el listado con los docentes disponibles		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Fecha de la denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Date	Tipo de dato	Date
Valor por defecto	Fecha denuncia	Tooltip	No aplica
Descripción*	Se muestra el listado		

Rango de valores permitidos	No aplica
Observaciones*	El sistema toma la fecha en el momento de la realización de la denuncia.

Agregar docente

En la vista de agregar docente (Ilustración 34), el usuario administrador puede agregar usuarios con el perfil de docentes diligenciado los campos de: Nombres (Aquí colocaremos nombres y apellidos) Teléfono y correo de este a petición del cliente.

También cuenta con el estado; Activo o Inactivo, según sea el caso el botón Registrar para confirmar los datos ingresados y el botón de Nuevo para seguir agregando Docentes.

Ilustración 34. Asignar caso a Docente, fuente autor

Tabla 30. Asignar caso a Docente, formato REQ_VIEW_V1

Asignar caso a Docente			
Campo*:	Asignar incidencias		
Longitud	Longitud de la denuncia	Obligatorio*	Si
Tipo control*	Combo Box	Tipo de dato	Texto
Valor por defecto	----	Tooltip	No aplica
Descripción*	Listado de docentes para asignar		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Agregar administrador

Un usuario administrador puede agregar a otros administradores. Para ello deberá registrar la contraseña de usuario por seguridad antes de otorgar tal permiso como lo muestra la ilustración 35. Una vez realizado esto, podemos ingresar los datos del nuevo administrador.

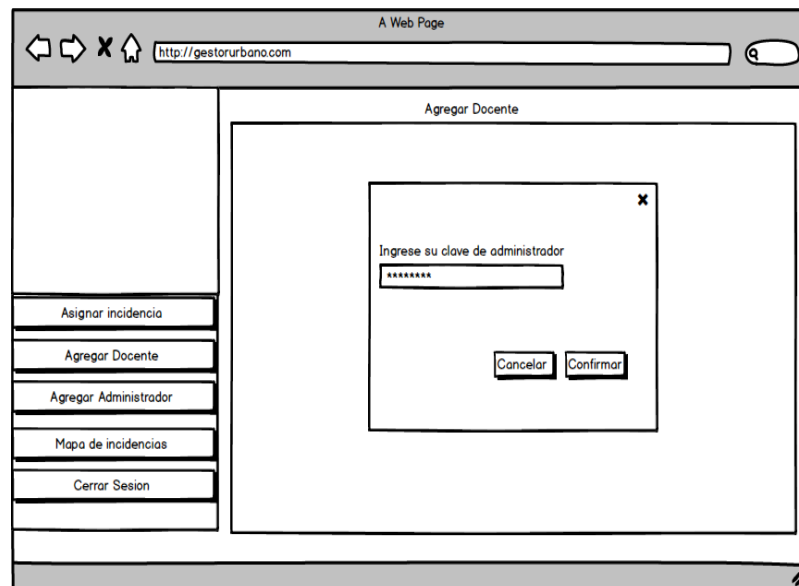


Ilustración 35. Solicitud de acceso administrador, fuente autor

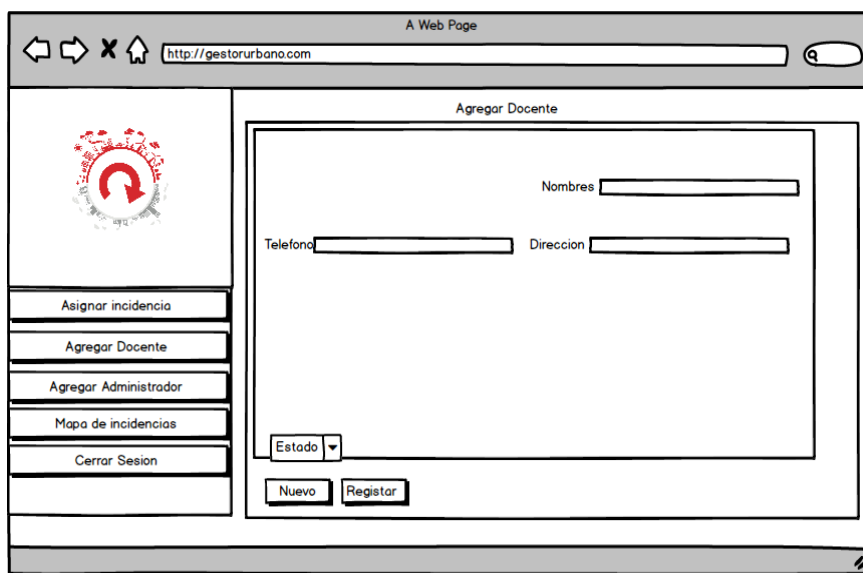


Ilustración 36. Agregar administrador, fuente autor

Tabla 31, Agregar administrador, formato REQ_VIEW_V1

Agregar administrador			
Campo*:	Nombres		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	texto	Tipo de dato	texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Nombres y apellidos del docente		
Rango de valores permitidos	Texto		
Observaciones*	No aplica		
Campo*:	Teléfono		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Text field	Tipo de dato	Numérico
Valor por defecto	No aplica	Tooltip	Si
Descripción*	Teléfono del docente		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Dirección		
Longitud	No aplica	Obligatorio*	si
Tipo control*	Texto	Tipo de dato	texto
Valor por defecto	No aplica	Tooltip	No aplica

Descripción*	Dirección de docente		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Nuevo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Botón	Tipo de dato	No aplica
Valor por defecto	Nuevo	Tooltip	No aplica
Descripción*	Botón para agregar nuevo usuario Administrador		
Rango de valores permitidos	Bolean		
Observaciones*	Se requiere contraseña de administrador antes de agregar user Admin		
Campo*:	Agregar		
Longitud	No aplica	Obligatorio*	No aplica
Tipo control*	Botón	Tipo de dato	No aplica
Valor por defecto	Agregar	Tooltip	No aplica
Descripción*	Agrega el Administrador diligenciado		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

Generar Reporte

En la ilustración 37 se muestra el resultado al generar el reporte en formato PDF, el cual es la denuncia detallada con todos los campos legales y estandarizados por el consultorio jurídico. Los datos diligenciados por el usuario ciudadano en la parte móvil, con cada una de las respuestas positivas del cuestionario de la denuncia, además de los datos del ciudadano como Nombres, Apellidos, número de documento, Lugar de expedición, Dirección donde se genera la denuncia y aspectos a intervenir.

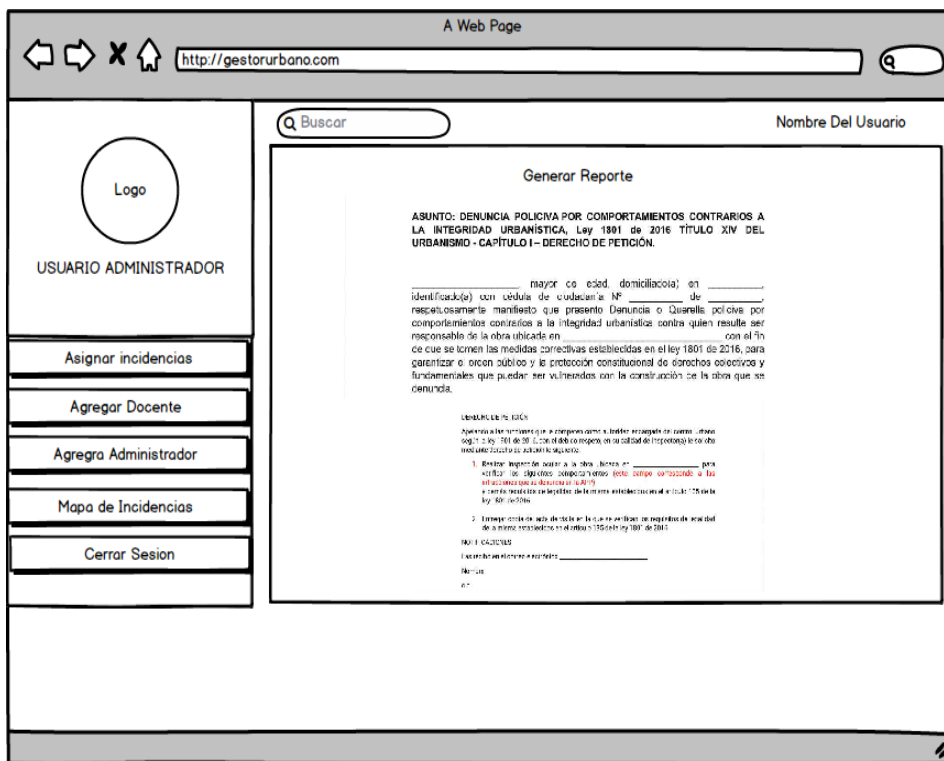


Ilustración 37. Generar Reporte, fuente autor

Tabla 32, Generar Reporte, formato REQ_VIEW_V1

Generar Reporte			
Campo*:	Generar Reporte		
Longitud	No aplica	Obligatorio*	No
Tipo control*	PDF	Tipo de dato	PDF
Valor por defecto	Formato denuncia	Tooltip	No aplica
Descripción*	Muestra el formato con la denuncia realizada, a la vez de traer los datos diligenciados por el usuario ciudadano.		
Rango de valores permitidos	PDF		
Observaciones*	No aplica		



Mapa de incidencias

Por último, tenemos el mapa de incidencias (Ilustración 38), en el cual podemos visualizar de una forma dinámica y agradable a primera impresión, con las zonas donde se han generado incidencias, localizadas con un icono predeterminado y su estado. Además de poder visualizar información relevante de quien y cuando realizó la denuncia.

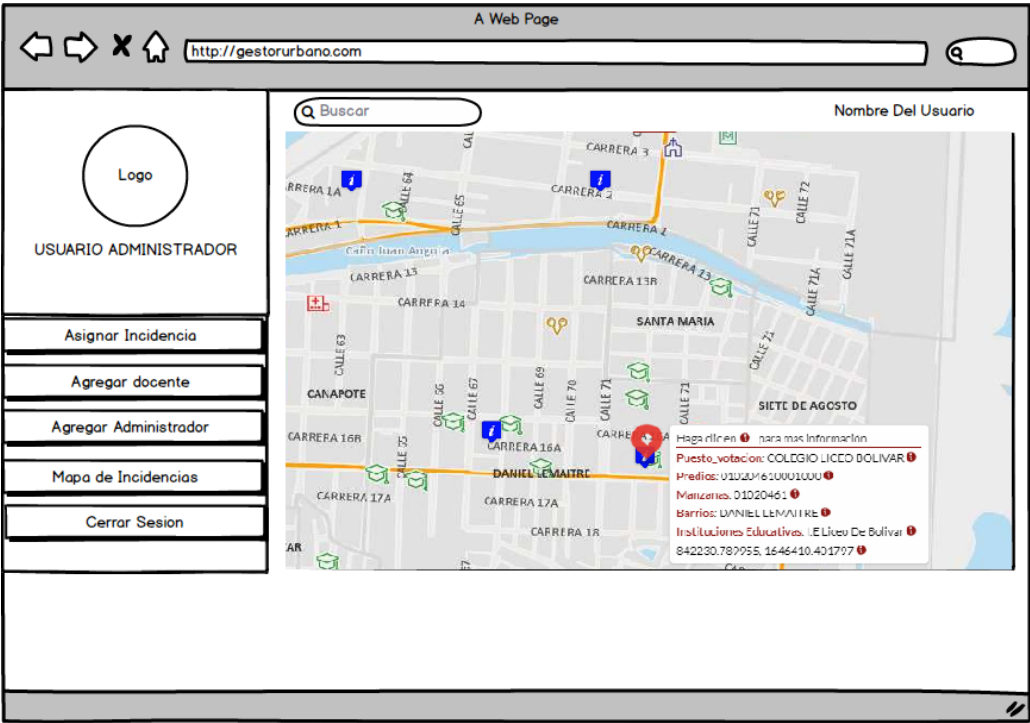


Ilustración 38. Mapa de incidencias, fuente autor

Tabla 33. Mapa de incidencias, formato REQ_VIEW_V1

Mapa de incidencias			
Campo*:	Mapa de denuncias		
Longitud	No aplica	Obligatorio*	No
Tipo control*	API Google Maps	Tipo de dato	API

Valor por defecto	Mapa de Cartagena	Tooltip	No aplica
Descripción*	Se muestra el mapa de la ciudad con los puntos donde se han generado las denuncias y se muestra información relevante con el estado de estas para su gestión		
Rango de valores permitidos			
Observaciones*	No aplica		

Página Principal

Esta vista de inicio es la ventana principal de la plataforma en donde el usuario Docente encontrará a disposición cada uno de los botones los cuales generan una acción específica, además de ofrecer una interfaz amigable y de uso intuitivo para una mejor experiencia de usuario.

Encontramos el logo del aplicativo en el lado izquierdo al igual que el rol del usuario. En la parte superior derecha encontramos el nombre del usuario logueado y la disposición de botones alineados a la izquierda tal y como lo ilustra la Ilustración 39

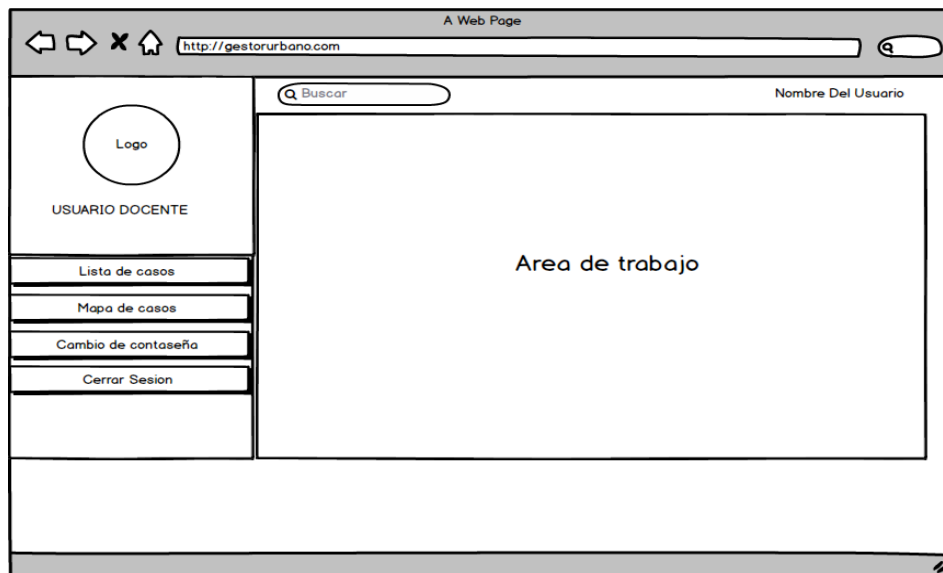


Ilustración 39. Página principal, fuente autor

Lista de casos

En la lista de casos, el docente podrá visualizar todas las denuncias que hayan sido asignadas por el administrador de la plataforma. Estas contienen un id el cual indica como número único del caso. Además de esto el usuario docente también puede consultar todos los datos relevantes de la denuncia, así como los datos del usuario quien la realiza. En la ilustración 40, podemos apreciar la distribución de los campos.

El estado de la denuncia puede ser modificado a medida que avanza el trámite de esta.

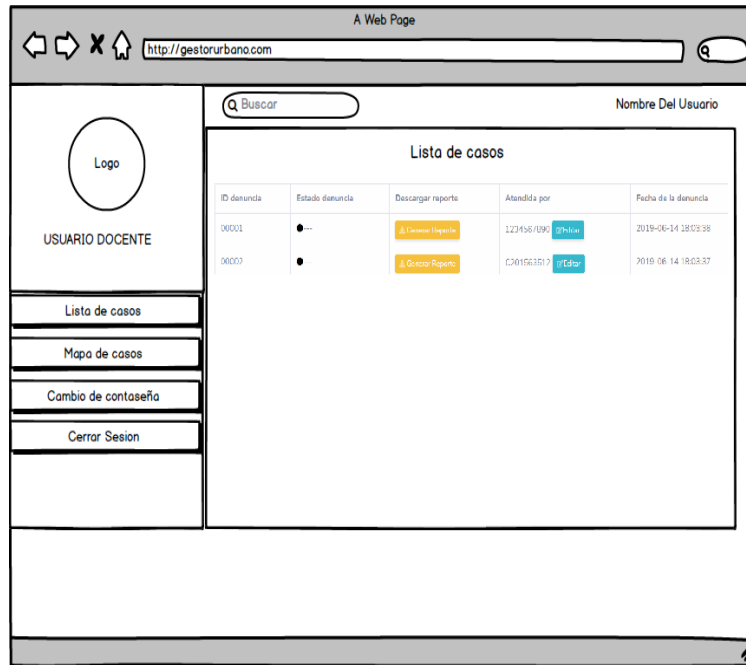


Ilustración 40. Lista de casos, fuente autor

Tabla 34. Lista de casos, formato REQ_VIEW_V1

Lista de casos			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	lg	No aplica
Descripción*	Logo de la institución		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Id denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Consecutivo de la denuncia		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Estado denuncia		

Longitud	No aplica	Obligatorio*	Si
Tipo control*	Botón	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Muestra el estado de la denuncia por un color en específico		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Descargar reporte		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	PDF	Tipo de dato	Texto
Valor por defecto	Botón	Tooltip	No aplica
Descripción*	Muestra la denuncia en formato PDF con todos los datos		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Atendido por		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo Box	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Se muestra el listado con los docentes disponibles		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Fecha de la denuncia		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Date	Tipo de dato	Date
Valor por defecto	Fecha denuncia	Tooltip	No aplica
Descripción*	Se muestra el listado		
Rango de valores permitidos	No aplica		
Observaciones*	El sistema toma la fecha en el momento de la realización de la denuncia.		

Notas del caso

En esta vista (ilustración 41), se muestran los campos para la gestión de la denuncia y agregar las respectivas notas a medida que el caso va avanzando. El estado es un

combo box que despliega cada uno de los cambios que una denuncia puede tener. Cada estado tiene un color característico que indica el avance que esta ha tenido. En caso de que la denuncia no sea viable el estado será de color rojo y se le notificará al usuario ciudadano la razón por la cual la denuncia no haya sido viable.

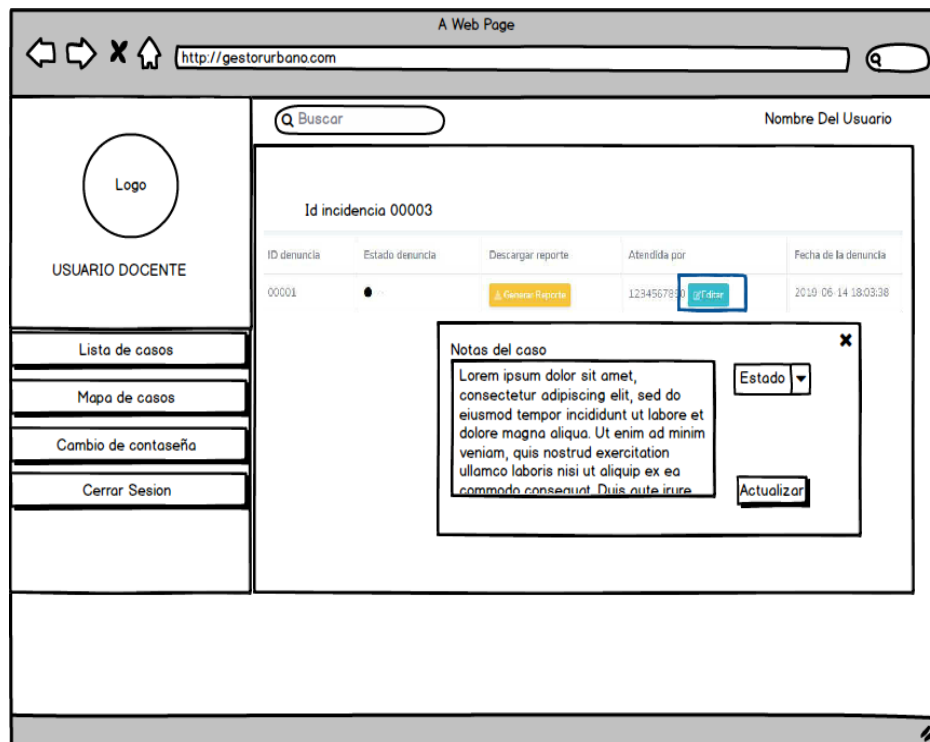


Ilustración 41. Notas del caso, fuente autor

Tabla 35. Notas del caso, formato REQ_VIEW_V1

Notas del caso			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Logo de la empresa		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Número de caso		

Longitud	No aplica	Obligatorio*	Si
Tipo control*	No aplica	Tipo de dato	Numérico entero
Valor por defecto	Número de caso	Tooltip	No aplica
Descripción*	Número de caso generado automático por el sistema		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Notas del caso		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para agregar notas relacionadas con la gestión de la denuncia por parte del docente.		
Rango de valores permitidos	No aplica		
Observaciones*	Estas denuncias serán visibles al ciudadano para su consulta.		
Campo*:	Estado		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo de búsqueda	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Lista de estados para la selección del usuario (Iniciado, En curso, Finalizado, Rechazado)		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Actualizar		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Botón de radio	Tipo de dato	Texto
Valor por defecto	PDF	Tooltip	No aplica
Descripción*	Botón para actualizar y guardar los cambios en base de datos		
Rango de valores permitidos	No aplica		
Observaciones*	Se muestra mensaje de confirmación al guardar.		

Mapa de incidencias

El mapa de incidencias muestra de forma dinámica las denuncias generadas ubicándose con un color específico y un icono el cual la identifica. Este icono al presionarlo muestra detalles relevantes de la denuncia. Esto hace que la interfaz sea más amigable dando una buena experiencia de usuario (ver ilustración 42).

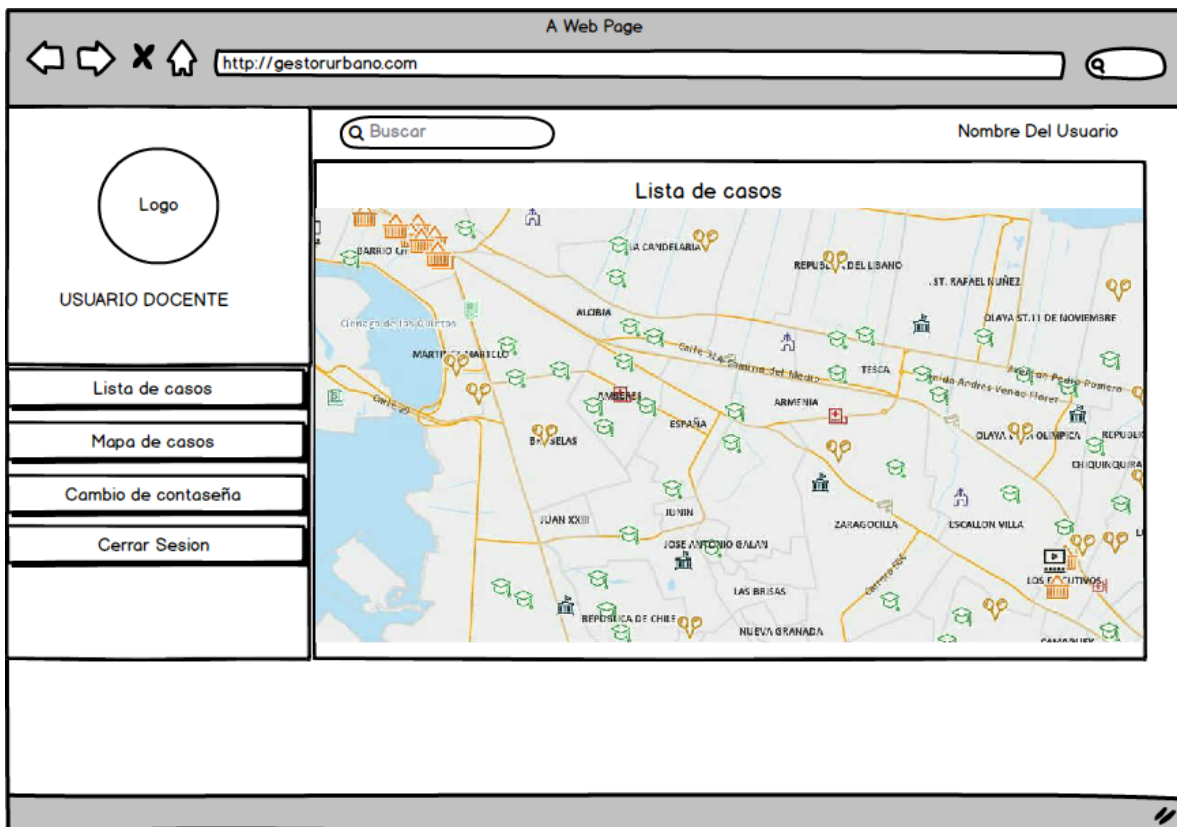


Ilustración 42. Mapa de denuncias, fuente autor

Tabla 36. Mapa de denuncias, formato REQ_VIEW_V1

Mapa de denuncias			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No

Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Logo de la empresa		
Rango de valores permitidos	JPEG,BMP,PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Numero de caso		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	No aplica	Tipo de dato	Numérico entero
Valor por defecto	Numero de caso	Tooltip	No aplica
Descripción*	Numero de caso generado automático por el sistema		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Notas del caso		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Texto
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Campo para agregar notas relacionadas con la gestión de la denuncia por parte del docente.		
Rango de valores permitidos	No aplica		
Observaciones*	Estas denuncias serán visibles al ciudadano para su consulta.		
Campo*:	Estado		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Combo de búsqueda	Tipo de dato	No aplica
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Lista de estados para la selección del usuario (Iniciado, En curso, Finalizado, Rechazado)		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Actualizar		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Botón de radio	Tipo de dato	Texto
Valor por defecto	PDF	Tooltip	No aplica

Descripción*	Botón para actualizar y guardar los cambios en base de datos
Rango de valores permitidos	No aplica
Observaciones*	Se muestra mensaje de confirmación al guardar.

Cambio de contraseña

El cambio de contraseña es muy importante por temas de seguridad. En la Ilustración 43 podemos ver que el usuario docente puede realizar el cambio de la contraseña si así lo desea. Para eso deberá colocar su contraseña actual y posteriormente ingresar la nueva. Al presionar el botón de guardar el sistema le mostrará un mensaje de confirmación efectuando los cambios.

Ilustración 43. Cambio de contraseña, Fuente autor

Tabla 37. Cambio de contraseña, formato REQ_VIEW_V1

Cambio de contraseña			
Campo*:	Logo		
Longitud	No aplica	Obligatorio*	No
Tipo control*	Imagen	Tipo de dato	Imagen
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Logo de la empresa		
Rango de valores permitidos	JPEG, BMP, PNG		
Observaciones*	Tamaño máximo 100KB		
Campo*:	Contraseña Anterior		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	*****	Tooltip	No aplica
Descripción*	Contraseña actual del usuario		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		
Campo*:	Nueva contraseña		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Nueva contraseña		
Rango de valores permitidos	No aplica		
Observaciones*	La contraseña puede incluir Mayúsculas y otros caracteres.		
Campo*:	Confirmar contraseña		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Cuadro de texto	Tipo de dato	Alfanumérico
Valor por defecto	No aplica	Tooltip	No aplica
Descripción*	Confirmar contraseña		
Rango de valores permitidos	No aplica		
Observaciones*	La contraseña puede incluir Mayúsculas y otros caracteres.		
Campo*:	Guardar		
Longitud	No aplica	Obligatorio*	Si
Tipo control*	Botón de radio	Tipo de dato	No aplica

Valor por defecto	Guardar	Tooltip	No aplica
Descripción*	Botón para la acción de guardar los cambios		
Rango de valores permitidos	No aplica		
Observaciones*	No aplica		

4. DESARROLLO DEL PROTOTIPO

Este capítulo detalla el desarrollo del software Gestor Urbano cuya principal función es generar y gestionar denuncias de control urbano de obras en construcción. Este fue diseñado utilizando la notación gráfica BPMN (modelo y notación de procesos de negocio), también se detalla el desarrollo de los tres módulos principales de la aplicación los cuales son: Administrador, Ciudadano y Docente; Utilizando la metodología de Desarrollo Rápido RAD. En este capítulo también se describen las métricas de desarrollo y ciertas características relevantes relacionadas a los modelos implementados para la obtención del producto final.

4.1 Tecnologías y lenguaje

Se utilizó la versión 7.2.7 de PHP (acrónimo recursivo de PHP: Hypertext Reprocesar) ya que este es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php y ?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP. [9]

Para la base de datos de este proyecto, se utilizó MySQL el cual es un sistema de gestión de base de datos relacional (RDBMS) de código abierto, basado en lenguaje de

consulta estructurado (SQL). Se ejecuta en prácticamente todas las plataformas, incluyendo Linux, UNIX y Windows. A pesar de que se puede utilizar en una amplia gama de aplicaciones, se asocia más con las aplicaciones basadas en la web y la publicación en línea y es un componente importante de una pila empresarial de código abierto llamado LAMP. LAMP es una plataforma de desarrollo web que utiliza Linux como sistema operativo, Apache como servidor web, MySQL como sistema de gestión de base de datos relacional y PHP como lenguaje de programación orientado a objetos (a veces, Perl o Python se utiliza en lugar de PHP). [10]

Se utilizó la quinta versión de HTML5 (HyperText Markup Language, versión 5). Esta nueva versión (aún en desarrollo), y en conjunto con CSS3, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándose así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, sino que incorpora muchas características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web (mediante los APIs).

Para los estilos y diseño, se utilizó la última evolución del lenguaje de las Hojas de Estilo en Cascada CSS (Cascading Style Sheets), ya que trae consigo muchas novedades altamente esperadas, como las esquinas redondeadas, sombras, gradientes, transiciones o animaciones, y nuevos layouts como multi-columnas, cajas flexibles o maquetas de diseño en cuadrícula (grid layouts).

JavaScript

Aprovechando que JavaScript es un lenguaje de programación que se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web, se utilizó para mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas entre otras.

Bootstrap

Este framework fue utilizado con el objetivo de facilitar el diseño web. Permitiendo crear nuestro proyecto web de forma sencilla y de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien. Es Open Source o código abierto, por lo que lo podemos usar de forma gratuita y sin restricciones.

Modelo de desarrollo: Para el desarrollo de la aplicación gestor urbano se efectuó el Modelo de Desarrollo Rápido de Aplicaciones (RAD), dicho modelo fue escogido ya que su enfoque va dirigido a las fases de modelado de gestión, modelado de datos, modelado de proceso, generación de aplicaciones y pruebas de entrega, esto junto con la metodología CMMI (Capability Maturity Model Integration).

Métricas de desarrollo: En el desarrollo del proyecto Gestor Urbano se aplicaron técnicas de ingeniería de software como lo son procesos de implementación de cambios, definición de procesos, Basado el swbook (Software Engineering Body of Knowledge) el cual se define como una guía al conocimiento presente en el área de la Ingeniería del Software.

También se tuvieron en cuenta las reglas estándares para la iniciación de variables en php de acuerdo al Manual de este, al igual que se siguieron lineamientos estándares para las sentencias SQL y la codificación en java script. Estas métricas pueden ser consultadas en el anexo 004

Desarrollo en PHP

Inicialización de variables

Aunque no es necesario, se recomienda inicializar las variables en PHP para reducir el tiempo de ejecución. Las variables no inicializadas tienen un valor predeterminado de acuerdo con su tipo (las booleanas se asumen como FALSE, los enteros y flotantes como cero, las cadenas se establecen como una cadena vacía y las matrices se convierten en un array vacío).

A continuación, se muestran ejemplos que implementan las reglas de construcción según manual PHP

Ejemplo de cómo aplicar estándar PHP en la iniciación de variables:

Función eval()

```
//Este código ejecuta $string como si fuera PHP.
```

```
//Por ejemplo:
```

```
<?php  
$name = 'Chris';  
$string = 'echo "Hello, $name"';  
eval($string);  
?>
```

```
//lo que es equivalente al código siguiente:
```

```
<?php  
$name = 'Chris';  
echo "Hello, $name";  
?>
```

```
<?php  
$name = $_GET['name'];  
eval($name);  
?>.
```

En el siguiente fragmento de código podemos apreciar la estructura de variables utilizada en el proyecto Gestor Urbano, aplicado sobre código Backend en el registro de usuario:

```
//Se inicializa el código php
```

<? ph p	
	require ("../unilab_clases/login_registration_Class.php");
	\$register = new login_registration();
	if(isset(\$_POST['Nombre']) && isset(\$_POST['Direccion']) && isset(\$_POST['Cedula'])
	&& isset(\$_POST['ExpedicionCedula']) && isset(\$_POST['Telefono']) && isset(\$_POST['Nombre_usuario'])
	&& isset(\$_POST['Email']) && isset(\$_POST['Clave']) { \$Nombre = \$_POST['Nombre']; \$Direccion = \$_POST['Direccion']; \$Cedula = \$_POST['Cedula']; \$Telefono = \$_POST['Telefono']; \$Nombre_usuario = \$_POST['Nombre_usuario']; \$Email = \$_POST['Email']; \$Clave = \$_POST['Clave']; \$ExpedicionCedula = \$_POST['ExpedicionCedula'] ;
	// Utilizando hash md5 para encriptación de la clave de usuario \$hash_Clave = md5(\$Clave,false);
	\$operation = \$register->registerUser(\$Nombre,\$Direccion,\$Cedula,\$ExpedicionCedula,\$Telefono,\$Nombre_usuario,\$Email,\$hash_Clave); echo(\$operation);
	}else{
	echo "consulta incorrecta"; }?>

Clases

Los nombres de clase deben ser sustantivos descriptivos en * PascalCase * y tan cortos como posible. Cada palabra en el nombre de la clase debe comenzar con una letra mayúscula, sin subrayar delimitadores. El nombre de la clase debe tener el prefijo nombre del "conjunto principal" (por ejemplo, el nombre de la extensión) si no hay espacios de nombres son usados. Las abreviaturas y acrónimos, así como los inicialismos deben ser evitados siempre que sea posible, a menos que sean mucho más utilizados que la forma larga (por ejemplo, HTTP o URL). Las abreviaturas comienzan con mayúscula seguido de letras minúsculas, mientras que los acrónimos y los inicialismos están escritos de acuerdo con su notación estándar. El uso de siglas e

inicialismos es no está permitido si no son ampliamente adoptados y reconocidos como tales. En la siguiente figura se muestra la aplicación de estas reglas en el proyecto gestor urbano. Para mayor detalle consultar anexo (004)

```
<?php
/**
 *
 */
class Teachers{

    private $conexion=false;
    function __construct($datos_conexion = false)
    {
        if(!$datos_conexion){
            $server="54.39.133.88";
            $username='gestorur_unilab';
            $password='unilab2019.';
            $db='gestorur_unilab_db';

            $this->conexion= new mysqli($server,$username,$password,$db);
            if($this->conexion->connect_errno){
                echo($this->conexion->connect_error);
            }else{
                $this->conexion->set_charset("utf8");
            }
        }
    }
}
```

Funciones y métodos

Los nombres de funciones para funciones de nivel de usuario deben incluirse en `PHP_FUNCTION ()` macro. Deben estar en minúsculas, con palabras de subrayado delimitado, con cuidado para minimizar el recuento de letras. Las abreviaturas no deben usarse cuando disminuyen en gran medida la legibilidad del nombre de la función. Para mayor detalle consultar anexo (004)

Forma correcta:

```
function nombreMetodo(parámetros) {
    // Código del Método
}
```

...

Forma Incorrecta:

```
<?php
hw_GetObjectByQueryCollObj
pg_setclientencoding
jf_n_s_i
```

...

En la siguiente figura se muestra la aplicación de estas reglas en el proyecto gestor urbano. Para mayor detalle consultar anexo D, Manual Técnico.

```
public function getAllDocList(){ //funcion que devuelve una lista para un select [OPTIONS] con los nombres de los docentes regist
    $consulta= "SELECT nombre from docentes";
    $query=$this->conexion->query($consulta);
    $lista = "<option>SELECCIONE</option>";

    while($datos = $query->fetch_assoc()){
        $lista = $lista."<option>".$datos['nombre']."</option>";
    }
    return $lista;
}

public function getIdOfDoc($nombre){ //funcion que devuelve el codigo del docente
    $consulta= "SELECT codigoDocente from docentes WHERE nombre='".$nombre."'";
    $query=$this->conexion->query($consulta);

    $codigo = $query->fetch_assoc();

    return $codigo;
}

public function getNameOfIdDoc($id){ //funcion que devuelve el nombre del docente
    $consulta= "SELECT nombre from docentes WHERE codigoDocente='".$id."'";
    $query=$this->conexion->query($consulta);

    $res = $query->fetch_assoc();
    $nombre = $res['nombre'];
    return $nombre;
}
```

SQL

Las convenciones a la hora de codificar ayudan a la uniformidad y legibilidad del código fuente, facilitando su mantenimiento por terceros y eliminando la dependencia del desarrollador.

Formato del código

El código SQL generado debe estar correctamente formateado siguiendo normas de indentación, alineación, bloques, etc. ya sea mediante el uso de una herramienta automática de formateo o de forma manual.

Estructura de las tablas

Los nombres de las tablas de la base de datos están escritos en singular y descriptivo, al igual que el nombre de los campos/ columnas también se encuentran en singular. A continuación, se muestra ejemplo de la estructura de la tabla:

NOMBRE_TABLA

idDocente

codigoDocente

especialidadDocente

Utilización de Sentencias:

Las distintas cláusulas (INTO, FROM, WHERE, ORDER BY, START WITH, CONNECT, HAVING, GROUP BY, etc.) y los operadores lógicos irán siempre al comienzo de una nueva línea.

Sentencias Insert / Delete / Update

Las distintas cláusulas en una nueva línea (WHERE, SET, VALUES), de forma análoga a la sentencia SELECT

```
INSERT INTO `administrador` (`nombre`, `direccion`, `cedula`, `lugarExpedicion`, `telefono`, `email`, `password`, `idAdministrador`)
VALUES ('brayan bravo silva', 'chile M 3 L6', '1047411410', 'cartagena', '6620050', 'adminbrayan@gtr.com', '25f9e794323b453885f5181f1b624d0b', 'pfiup');
```

La siguiente imagen ilustra la utilización de la sentencia CREATE TABLE. Aplicando buenas prácticas

```
CREATE TABLE `CUESTIONARIO` (
  `idCuestionario` varchar(20) CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL,
  `descripcionCuestionario` text CHARACTER SET utf8 COLLATE utf8_spanish_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.2 Módulos del sistema

Gestor urbano cuenta con tres módulos principales tales como Ciudadano, Administrador y docentes los cuales se detallan a continuación:

Modulo ciudadano

Este módulo fue desarrollado pensando en la seguridad de la información del usuario, además de que su interfaz se amigable y lo suficientemente intuitiva para mayor facilidad a la hora de usarla. Se utilizó el algoritmo de cifrado MD5, ya que este algoritmo no es reversible y normalmente es imposible encontrar la palabra original del MD5.

En este módulo se explica la transición entre el diagrama de componentes hasta llegar a la vista FrontEnd final dando como respuesta a los requerimientos del sistema: RF001, RF002, RF003, RF004, RF005, RF006, RF007, RF008.

En la ilustración 44 se presentan los componentes del módulo ciudadano y las vistas relacionadas al mismo. Desde la autenticación, vista crear denuncia, consultar denuncia, consultar información general y consultar las actualizaciones agregadas al caso (notas).

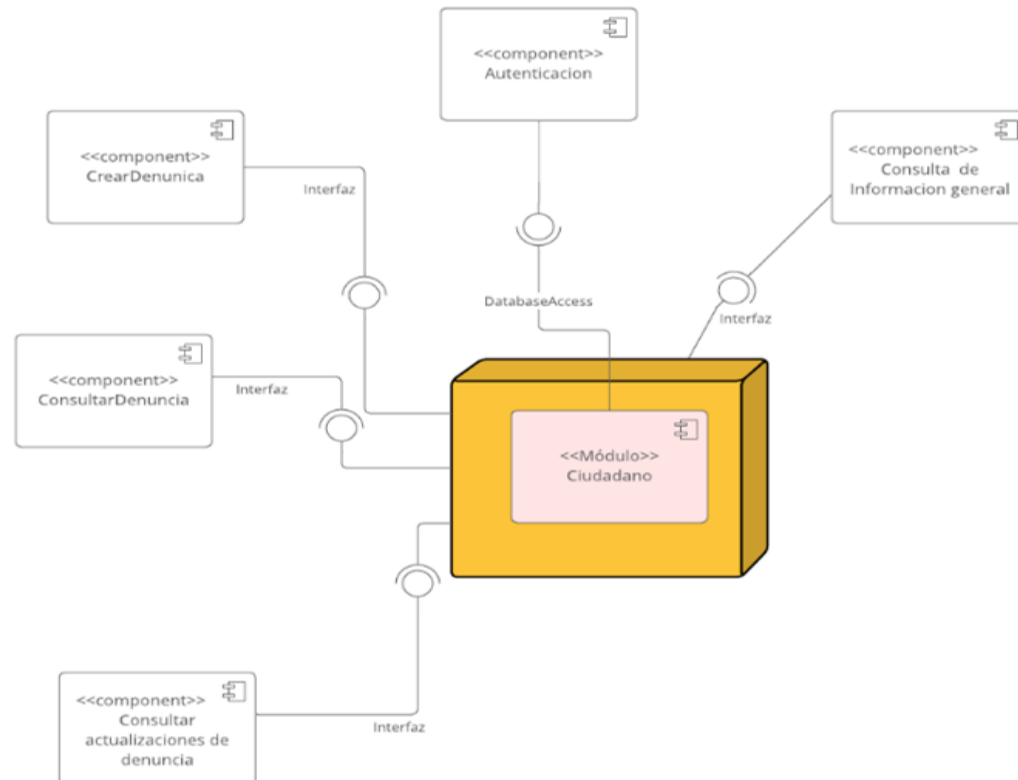


Ilustración 44. Diagrama de componentes módulo Ciudadano, Fuente Autor

Autenticación de usuario

La ilustración 45 describe el diseño en mockup de la interfaz de inicio de sesión, en esta se planteó la utilización de un combo box que desplegará las diferentes opciones según el actor,



Ilustración 45. Mockup Logging, Fuente autor



Ilustración 46. Vista Frontend login, Fuente autor

La autenticación de usuario, cuenta con: una lista desplegable para seleccionar el tipo de cuenta, ya sea Ciudadano, Administrador o Docente; un campo de texto para ingresar el correo electrónico su respectiva validación ; un campo para el ingreso de contraseña que cuenta con encriptación hash MD5; En caso de olvidar los datos de

acceso, el sistema cuenta con la opción de registro y recordar contraseña, la cual será enviada al correo electrónico del usuario en caso de solicitud de recuperación.

Para transformar el diseño del mockup en la interfaz construida (ilustración 46) fue necesario utilizar las etiquetas HTML `<Option >` la cual se usa para representar un ítem dentro de un select en la base de datos; la etiqueta `< label>` para correo electrónico y contraseña. Como Se puede apreciar no hubo variación en el diseño en comparación con la interfaz frontend descrita en la ilustración 46.

Se puede apreciar en el siguiente fragmento de código la estructura frontend de los labels antes mencionados, los diferentes inputs pertenecientes al `<form>` al igual que la utilización de la etiqueta placeholder para identificar la información de los campos correo y contraseña de la vista.

```
<option>Seleccione</option>
<option>Ciudadano</option>
<option>Administrador</option>
<option>Docente</option>
</select>
<label for="inputEmail">Correo electrónico</label>
<input type="email" class="form-control" id="inputEmail" placeholder="usuario@usuario.com" pattern="[A-Za-z0-9@
<label for="inputClave">Contraseña</label>
<input type="password" class="form-control" id="inputClave" placeholder="*****" pattern="[A-Za-z0-9]{4,15}">
<label id="consola"></label><br>
<button type="submit" id="btn-login" class="btn btn-danger">INGRESAR</button><br><br>
<label><a href="registro.php" class="labelLogin">Registrarse</a></label><br>
<label><a href="./passRecovery.php" class="labelLogin">¿olvidó su contraseña?</a></label><br>
</form>
```

Por otra parte, en el siguiente fragmento se describe la estructura backend utilizada para requerir los datos ingresados por el usuario utilizando el método POST, se importa la clase login registration class.php se implementa el hash md5 para el campo “clave” y se muestra la validación del usuario utilizando Switch Case.

```

require("../unilab_clases/login_registration_Class.php");
if(isset($_POST['cuenta']) && isset($_POST['email']) && isset($_POST['clave']))(
    $entry = new login_registration ();
    $email=$_POST['email'];
    $clave=$_POST['clave'];
    $cuenta=$_POST['cuenta'];
    $shash = md5($clave);
    // echo "$cuenta";
    switch ($cuenta) {
        case 'Ciudadano':
            if($entry->getUserWithPass($email,$shash)){
                echo true;
            }else{
                echo "Usuario o clave inválida";
            }
            break;
        case 'Administrador':
            if($entry->getAdminWithPass($email,$shash)){
                echo true;
            }else{

```

Para mayor detalle de la codificación del login, anexo D, Manual Técnico.

Registro de usuario

La ilustración 47 describe el diseño en mockup de la interfaz para el registro de usuario, en esta se planteó la utilización text box para la solicitud de los datos requeridos y un botón para registrar los datos.

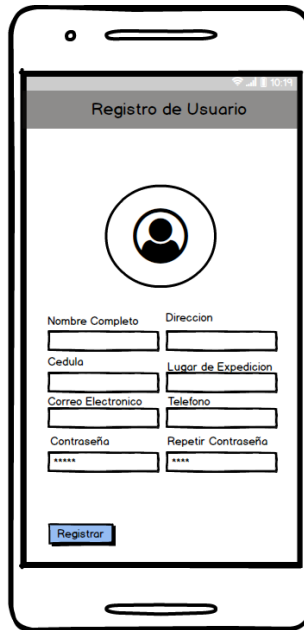


Ilustración 47. Mockup Registro de usuario, fuente autor

En el registro de usuarios el sistema permite al ciudadano ingresar sus datos en la aplicación para que estos sean almacenados en base de datos. Los datos solicitados son relevantes para las posteriores denuncias, ya que el documento final (la denuncia) toma estos datos del usuario y los presenta en el PDF.


GESTOR URBANO REGISTRO DE USUARIO

Nombre completo	Dirección	
<input type="text" value="Pablo Perez Paso"/>	<input type="text" value="Cra 1 No 4 CASA 2"/>	
Cedula	Lugar de Expedición	
<input type="text" value="1.222.333.444"/>	<input type="text" value="CARTAGENA-BOLIVAR"/>	
Nombre de usuario	Correo electrónico	Telefono
<input type="text" value="Pablo0123"/>	<input type="text" value="usuario@usuario.com"/>	<input type="text" value="300 000 0000"/>
Contraseña	Repetir contraseña	
<input type="password" value="*****"/>	<input type="password" value="*****"/>	

REGISTRAR

Ilustración 48. Vista frontend registro de usuario, fuente autor

En la ilustración 48 se observa que la vista cuenta con los siguientes campos: Nombre completo, Dirección, Cédula, Lugar de expedición, Correo electrónico, teléfono, contraseña y confirmación de contraseña; en el diseño frontend estos campos fueron creados utilizando las etiquetas h1 para título de la vista, La identificación de los “inputs” con sus respectivos “id” cada label se encuentra en una etiqueta div para hacer más fácil el manejo de los estilos; obsérvese que cada campo de texto contiene su respectivo placeholder para orientar al usuario, estos se pueden observar en el siguiente fragmento de código frontend:

```
<div id="contenedorGeneral" class="bg-dark">
  <div id="contenedorFormulario" class="shadow-lg p-3 mb-5 bg-white rounded">
    <form id="f">
      
      <h1 class="h3 mb-3 font-weight-normal">GESTOR URBANO REGISTRO DE USUARIO</h1><br>
      <div class="form-row">
        <div class="col">
          <label for="inputNombre"><b>Nombre completo</b></label><br>
          <input type="text" class="form-control" id="inputNombre" placeholder="Pablo Perez Paso">
        </div>
        <div class="col">
          <label for="inputDireccion"><b>Direccion</b></label>
          <input type="text" class="form-control" id="inputDireccion" placeholder="Cra 1 No 4 CASA 2" patter
        </div>
      </div>
    </div>
    <div class="form-row">
      <div class="col">
        <label for="inputCedula"><b>Cedula</b></label><br>
        <input type="text" class="form-control" id="inputCedula" placeholder="1.222.333.444">
      </div>
    </div>
  </div>
</div>
```

Por otra parte en el siguiente fragmento se describe la estructura backend utilizada para requerir los datos ingresados por el usuario utilizando el método POST; se inicia el código importando la clase login registration y luego la estructura de control “if” como condición para la verificación de las variables definidas y se finaliza con la operación “registrar”.

```
<?php
require ("../unilab_clases/login_registration_Class.php");
$register = new login_registration();

if(isset($_POST['Nombre']) && isset($_POST['Direccion']) && isset($_POST['Cedula'])
    && isset($_POST['ExpedicionCedula']) && isset($_POST['Telefono']) && isset($_POST['Nombre_usuario'])
    && isset($_POST['Email']) && isset($_POST['Clave'])){

    $Nombre = $_POST['Nombre'];
    $Direccion = $_POST['Direccion'];
    $Cedula = $_POST['Cedula'];
    $Telefono = $_POST['Telefono'];
    $Nombre_usuario = $_POST['Nombre_usuario'];
    $Email = $_POST['Email'];
    $Clave = $_POST['Clave'];
    $ExpedicionCedula = $_POST['ExpedicionCedula'];
    $hash_Clave = md5($Clave,false);

    $operation = $register->registerUser($Nombre,$Direccion,$Cedula,$ExpedicionCedula,$Telefono,$Nombre_usuario,$Email,$hash_Clave);
    echo($operation);
}else{
```

Para mayor detalle de la codificación del registro de usuario, anexo D, Manual Técnico.

Vista de pantalla principal ciudadano

La ilustración 49, describe el diseño en mockup de la interfaz de la página principal, en esta se planteó la utilización de un menú situado en la parte izquierda de tal modo que el área central y lateral derecha quedará disponible como área de trabajo.

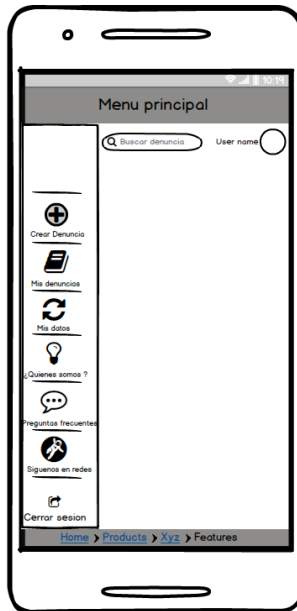


Ilustración 49. Mockup vista principal ciudadano, fuente autor

Para transformar el diseño del mockup en la interfaz construida (ilustración 50) se utilizaron las etiquetas `` de HTML como definición de los ítems del menú, tales como: Crear denuncia, Mapa de casos, actualizar datos cambio de contraseña y las opciones generales como “quienes somos” cada uno de estos despliega un modal donde se muestra la información correspondiente según los requisitos del sistema; la clase sidebar- divide es un componente de navegación de bootstrap la cual es ideal para el tipo de menú vertical.

A continuación de muestran la implementación de las etiquetas en la estructura frontend:

```

<li class="nav-item">
  <a class="nav-link" href="#" data-toggle="modal" data-target="#modalCambioClave">
    <i class="fas fa-fw fa-lock"></i>
    <span>Cambiar contraseña</span></a>
</li>
<li class="nav-item">
  <a class="nav-link" href="../login/cerrarSesion.php">
    <i class="fas fa-fw fa-sign-out-alt"></i>
    <span>Cerrar sesion</span></a>
</li>

<!-- Divider -->
<hr class="sidebar-divider d-none d-md-block">

<div class="sidebar-heading">
  Acerca de
</div>
<li class="nav-item">
  <a class="nav-link" href="../login/cerrarSesion.php">
    <i class="fas fa-fw fa-lightbulb"></i>
    <span>¿Quiénes somos?</span></a>

```

Para la vista de la pantalla principal o de inicio, se decidió colocar un menú bastante simplificado a la parte izquierda, este se puede ocultar al presionar el botón (<), además se realizó una variación en el diseño colocando un chat enlazado con la fan page de Facebook de la plataforma, el cual se encuentra de modo flotante en la parte inferior derecha, donde el ciudadano puede interactuar con los asesores de consultorio jurídico en tiempo real.

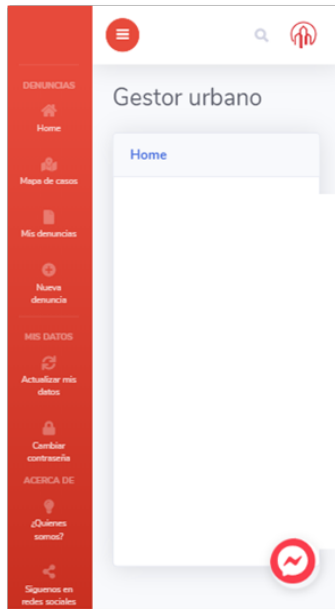


Ilustración 50. Frontend vista página principal ciudadano, fuente autor

En el siguiente fragmento de código se aprecia la estructura backend para la implementación plugin de chat Messenger que ofrece directamente Facebook. Para la utilización de este chat se debe contar con una cuenta de Facebook para empresas. En la ruta de configuración de la Fanpage -> Plataforma de Messenger podemos ejecutar un código Javascript colocándolo en la etiqueta <body > de nuestro código HTML.


```

<!-- Load Facebook SDK for JavaScript -->
<div id="fb-root"></div>
<script>
  window.fbAsyncInit = function() {
    FB.init({
      xfbml      : true,
      version    : 'v4.0'
    });
  };

  (function(d, s, id) {
    var js, fjs = d.getElementsByTagName(s)[0];
    if (d.getElementById(id)) return;
    js = d.createElement(s); js.id = id;
    js.src =
'https://connect.facebook.net/en_US/sdk/xfbml.customerchat.js';
    fjs.parentNode.insertBefore(js, fjs);
  })(document, 'script', 'facebook-jssdk');</script>

<!-- Your customer chat code -->
<div class="fb-customerchat"

```

Para mayor detalle de la codificación del Login consultar anexo D, Manual Técnico.

Diligenciamiento de formato de denuncia

En la ilustración 51 describe el diseño en mockup de la interfaz de nueva denuncia, en esta se planteó la utilización de un combo formato el cual se encuentra en HTML, este consta de un campo de texto para ingresar la cantidad de pisos, este campo solo acepta número, los demás elementos de las preguntas están en modo checkbox para marcar con sí o no.



Ilustración 51. Mockup diligenciar denuncia, fuente autor

Para transformar el diseño del mockup en la interfaz construida (ilustración 52) se creó una clase llamada Nueva Denuncia en la cual se visualiza un modal que muestra las preguntas contenidas en las etiquetas `<td>` la cual representa a una celda de datos de una tabla para el caso de cada pregunta, en el código a continuación se toma como ejemplo la pregunta “¿Cuántos pisos tiene la obra?” donde también se muestra la clase `class="fas fa-fw fa-info"` la cual se utiliza para colocar el signo de admiración para brindar más información acerca de la pregunta; el `input type="radio"` permite seleccionar con un clic las opciones de “Sí” o “No”.

```

<div class="modal fade NuevaDenuncia" id="md" tabindex="-1" role="dialog" aria-labelledby="myExtraLargeModallabel" aria-hidden="true">
  <div class="modal-dialog modal-xl">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <table class="table table-hover">
          <thead>
            <tr><th colspan="3" style="text-align:center;color:white;" class="bg-danger">Diligenciar Formato</th></tr>
          </thead>
          <tbody>
            <tr>
              <td>¿Cuántos pisos tiene la obra?
                <i class="fas fa-fw fa-info" onclick="info(0)" style="cursor:pointer;"></i>
              </td>
              <td colspan="2">
                <input type="number" id="inputAnswer1" class="form-control" value="0">
              </td>
            </tr>
          </tbody>
        </table>
      </div>
    </div>
  </div>
</div>

```

Agregar imágenes

Se agregó la opción de cargar tres imágenes a la denuncia, se utilizó la función `image verify` en donde se llama la url `leerImg` utilizando el método `POST` y luego se valida que efectivamente se hayan cargado las tres imágenes; el código se puede observar a continuación:

```

function imageVerify(){
  var data = new FormData();
  jQuery.each($('input[type=file]')[0].files, function(i, file) {
    data.append('file-'+i, file);
  });
  $.ajax({
    type: 'POST',
    url: "leerImg.php",
    data: data,
    cache: false,
    contentType: false,
    processData: false,
    method: 'POST',
    // Mostramos un mensaje con la respuesta de PHP
    success: function(data) {
      if(data == 3){
        $('#imageLoadResult').html("<i class='fas fa-fw fa-check-circle' onclick='info(8)'' style='cursor:pointer;color: gr
      }else{
        $('#imageLoadResult').html("<i class='fas fa-fw fa-exclamation-triangle' onclick='info(8)'' style='cursor:pointer;c

```

En el caso de la estructura Backend de la Ilustración 52, se utilizó la librería HTML2PDF; esta se caracteriza por ser una herramienta muy sencilla y útil para la tarea de posicionar elementos y darle estilos al PDF que es el resultado de la denuncia, mediante HTML y CSS. Como resultado se obtienen que las respuestas que fueron marcadas como “NO “en el formulario se posicionan en el PDF al igual que la dirección y los datos del ciudadano. Para la utilización de este script es muy importante incluir la librería de la siguiente manera: `require_once 'html2pdf_v4.03/html2pdf.class.php'`; luego se recoge el contenido de la vista con `require dir_`, luego en `$html2pdf = new Html2pdf` (parámetros) indicamos el tipo de hojas y caracteres y finalmente con `&html2pdf-> output` se genera el PDF con los datos del formulario esto se puede en el siguiente fragmento de código:

```
<?php
}
$content = ob_get_clean();
require __DIR__.'/vendor/autoload.php';

use Spipu\Html2Pdf\Html2Pdf;
try
{
    $html2pdf = new Html2Pdf('P', 'letter', 'es', true, 'UTF-8',3);
    $html2pdf->pdf->setDisplayMode('fullpage');
    $html2pdf->writeHTML($content, isset($_GET['vuehtml']));
    $html2pdf->Output('PDF-CF.pdf');
}
catch(HTML2PDF_exception $e) {
    echo $e;
    exit;
}

>>
```

Como resultado se obtiene que los datos ingresados en la vista 52 son presentados en formato PDF como denuncia final.

Diligenciar Formato

¿Cuántos pisos tiene la obra? 0

¿Tiene escaleras por fuera de la edificación? SI NO

¿Tiene garajes o parqueaderos? SI NO

¿Tiene la obra instalada la valla informativa de la licencia de construcción? SI NO

¿La obra se construye sobre espacio público? SI NO

¿La obra afecta la integridad del espacio público? (andén - calle) SI NO

¿La obra ocupa con material de construcción el andén, las vías o espacios públicos circundantes? SI NO

¿Representa la obra un peligro para el transeúnte? SI NO

¿Cuenta la obra con la debida protección para evitar la contaminación del sector donde se ubica? (Se refiere este punto a mallas o barreras de protección) SI NO

¿Desarrolla la obra actividades de construcción en la vía pública, afectando el entorno? (Eje. Preparación de mezclas en la vía pública) SI NO

Agregue 3 imagenes


▲

Escriba la dirección

Av. pedro de heredia No 12
 ID 3867959-75.49271639999999

Agregar Ubicación

Mapa
Satélite



Datos de mapa ©2019 Términos de uso

Enviar resultados

Volver al menu principal

Ilustración 52. Vista front nueva denuncia, fuente autor

La visualización en el mapa se realizó utilizando la API de Google llamada: Maps JavaScript API, gracias a que nos permite personalizar el mapa con contenido propio, con la ayuda de sus diferentes capas y estilos, control de eventos, servicios y librerías variadas, y su versatilidad a la hora de visualizarse en cualquier dispositivo, la geolocalización y el posicionamiento de marcas en el mapa. Al momento del ciudadano generar, la denuncia, el sistema fija las coordenadas localizando la marca en el punto donde esta se origina

La localización se captura a través de la función `getMyLocation` con la variable “`coor`” perteneciente a las coordenadas y luego se realiza la validación con la estructura de control “`if`” para mostrar la posición la siguiente manera:

```
function getMyLocation(){
    $("#lblLocation").html("espere...");
    var coor;
    if(navigator.geolocation){
        navigator.geolocation.getCurrentPosition(showPosition);
    }else{
        $("#lblLocation").html("su navegador no soporta esta funcion");
    }
}
```

Para el posicionamiento de la marca se utiliza el siguiente script

```
var map = new google.maps.Map(
    document.getElementById('map'), {zoom: 4, center: (ubicación en el mapa ), });
// La marca , posicionada en la ubicación en el mapa
var marker = new google.maps.Marker({position: (ubicación en el mapa ), map:
map});
}
</script>
<script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=ini
tMap">
</script>
```

De esta forma queda predeterminada la marca en el mapa como lo muestra la ilustración 56.

Para mayor detalle de la codificación del diligenciamiento de la denuncia, consultar anexo D, Manual Técnico.

10.3867959,-75.49271639999999

Agregar Ubicación



Enviar resultados

Volver al menú principal

Ilustración 53. Vista front nueva denuncia, fuente autor

Podemos observar una variación entre el mockup de la ilustración 51 y la vista front de la ilustración 52, ya que se omitió la opción de deseo “gestionar mi denuncia” por solicitud del cliente. En cambio, todas las denuncias son generadas y almacenadas para que el usuario ciudadano las pueda consultar desde su módulo al igual de que sean gestionadas por consultorio jurídico, además se implementó que cada imagen sea subida de forma individual.

Para mayor detalle de la codificación del funcionamiento de la API, anexo D, Manual Técnico.

Consultar denuncias

En la ilustración 54 se describe el diseño en mockup de la interfaz para la consulta de denuncias generadas; en esta, se utiliza una tabla que muestra la información correspondiente al id de la denuncia, estado, descargar reporte de denuncia y consultar las notas agregadas a la misma.

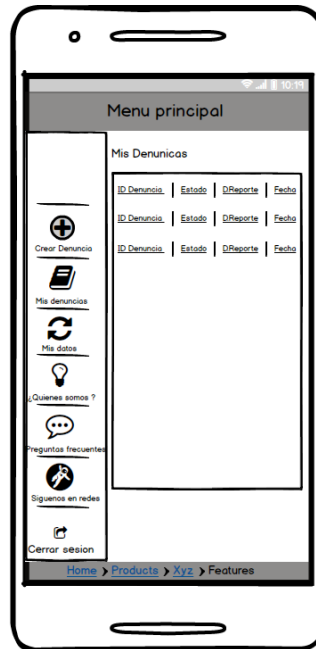


Ilustración 54. Mockup consultar denuncia, fuente autor

La estructura de la vista en la Ilustración 54, consta de la clase `table-responsive` posible estilizada con librería `datatable` de `bootstrap` en donde los identificadores de las columnas como `ID denuncia`, `Estado denuncia`, `Notas`, `Descargar reporte`, `Atendida por`, y `fecha de la denuncia` se encuentran dentro de la etiqueta HTML `<th>`, está define una celda como la cabecera de una columna en una tabla. A continuación, se muestra en el bloque de código:

```
<div id="screen_denuncia" class="table-responsive" style="display: none;">
  <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
    <thead>
      <tr>
        <th>ID denuncia</th>
        <th>Estado denuncia</th>
        <th>Notas</th>
        <th>Descargar reporte</th>
        <th>Atendida por</th>
        <th>Fecha de la denuncia</th>
      </tr>
    </thead>
    <tbody>
```


Para el caso del estado de la denuncia se definieron los siguientes: en estudio, en trámite, presentada, finalizada y rechazada. La estructura se definió utilizando la clase getColor sobre la variable estado quedando de la siguiente manera:

```
<?php
function getColor($estado){
    switch ($estado) {
        case 'EN ESTUDIO':
            return '#FFE526';
            break;
        case 'EN TRAMITE':
            return '#FF6200';
        case 'PRESENTADA':
            return '#0081EB';
        case 'FINALIZADA':
            return '#05C400';
        case 'RECHAZADA':
            return '#EB0800';
    }
}
```

El siguiente fragmento de código muestra la función getMyComplaints la cual se implementa para mostrar las denuncias en la vista.

```
public function getMyComplaints($idUserario,$filtro = false){
    switch ($filtro) {
        case 'All':
            $consultaDenuncias= "SELECT * from denuncia";
            break;

        default:
            $consultaDenuncias= "SELECT idDenuncia,fecha,estado,idDocente from denuncia WHERE idUsuario='$idUserario'";
            break;
    }
}
```

La imagen 55 ilustra el diseño final para la consulta de las denuncias realizadas, estas se encuentran en el botón del menú lateral izquierdo como “Mis denuncias “. Esta vista tiene como finalidad descargar la denuncia generada a través del botón “Generar Reporte “al igual que consultar las notas agregadas a la misma.

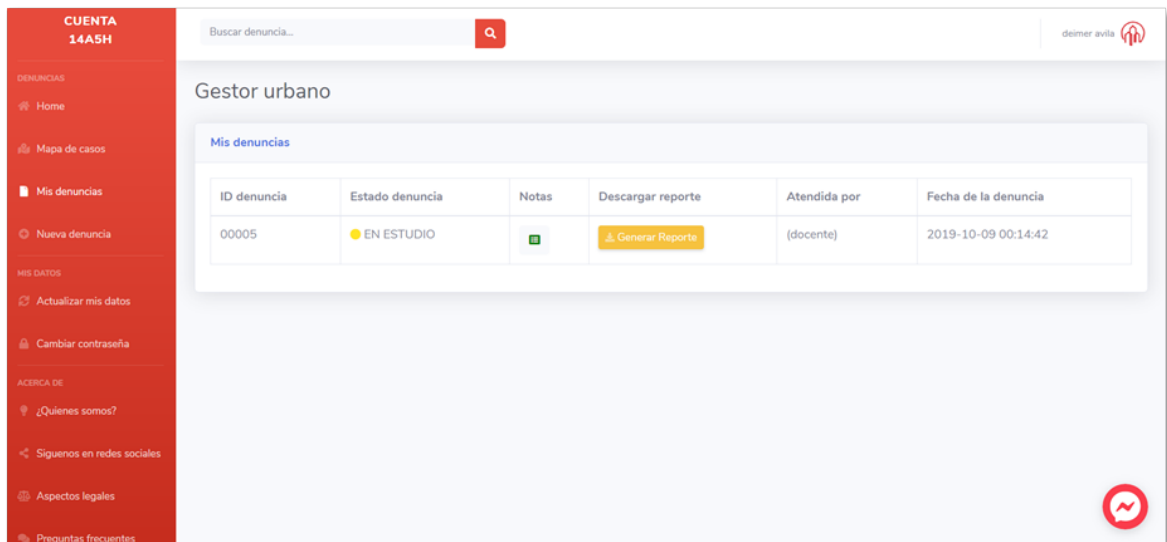


Ilustración 55. Vista frontend consultar denuncia, fuente autor

Para mayor detalle de la codificación de la consulta de denuncia, consultar anexo D, Manual Técnico.

Vista de información general

En la Ilustración 56 se describe el diseño en mockup de la interfaz para la consulta de la información general, se planteó colocar las opciones quienes somos, aspectos legales y preguntas frecuentes.

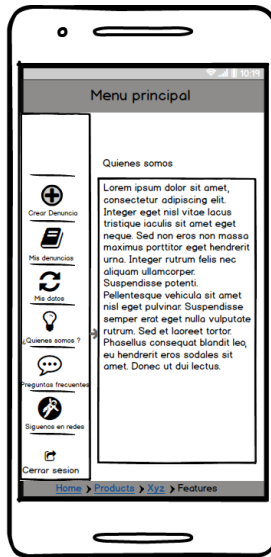


Ilustración 56. Vista de Información general, fuente autor

La estructura front de la vista de la información general consta de un modal perteneciente a la clase modal fade, esta clase permite crea un modo que deshabilita la ventana principal, pero la mantiene visible, con la ventana modal como una ventana secundaria frente a ella, dicha clase consta el id de modal- (nombre del modal), los div de lo componen, los estilos de la vista y el tipo de fuente.

```

<!-- Modal aspectos legales -->
<div class="modal fade" id="modal-quienes-somos" tabindex="-1" role="dialog"
  aria-labelledby="exampleModalScrollableTitle" aria-hidden="true">
  <div class="modal-dialog modal-dialog-scrollable" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title"><b><em>¿Quiénes somos?</em></b></h5>
        <button type="button" class="close" data-dismiss="modal" aria-label
          ="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
    </div>
  </div>

```

```

<div class="modal-body">
  Somos una App móvil creada como una estrategia de innovación social
  universitaria responsable aplicada al desarrollo urbanístico
  sostenible del distrito de Cartagena. Pretendemos dar solución a las
  irregularidades urbanísticas de forma fácil, viable e innovadora,
  trabajando de la mano con la plataforma Unilab, creando espacios de
  participación ciudadana y fortaleciendo el conocimiento de la norma.
</div>

```

Como resultado final tenemos la vista de la Ilustración 57 donde se muestra la un modal para las opciones informativas del sistema. Este cuenta con un apartado individual para quienes somos, información de redes sociales de la aplicación, aspectos legales y preguntas frecuentes.

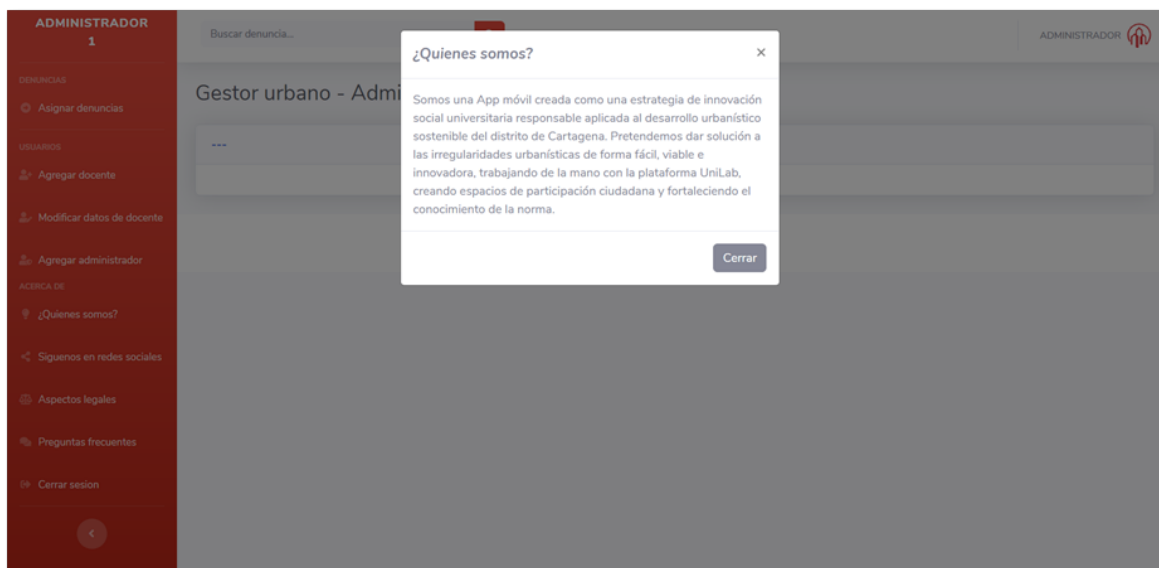


Ilustración 57. Vista Frontend de información general, fuente Autor

Para mayor detalle de la codificación de la información general, consultar anexo D, Manual Técnico.

Módulo administrador

En la ilustración 58 se muestran los componentes del módulo administrador y las vistas relacionadas al mismo se; describe la vista asignar denuncia, mapa de casos, Agregar preguntas al cuestionario, Timer y agregar usuarios al sistema.

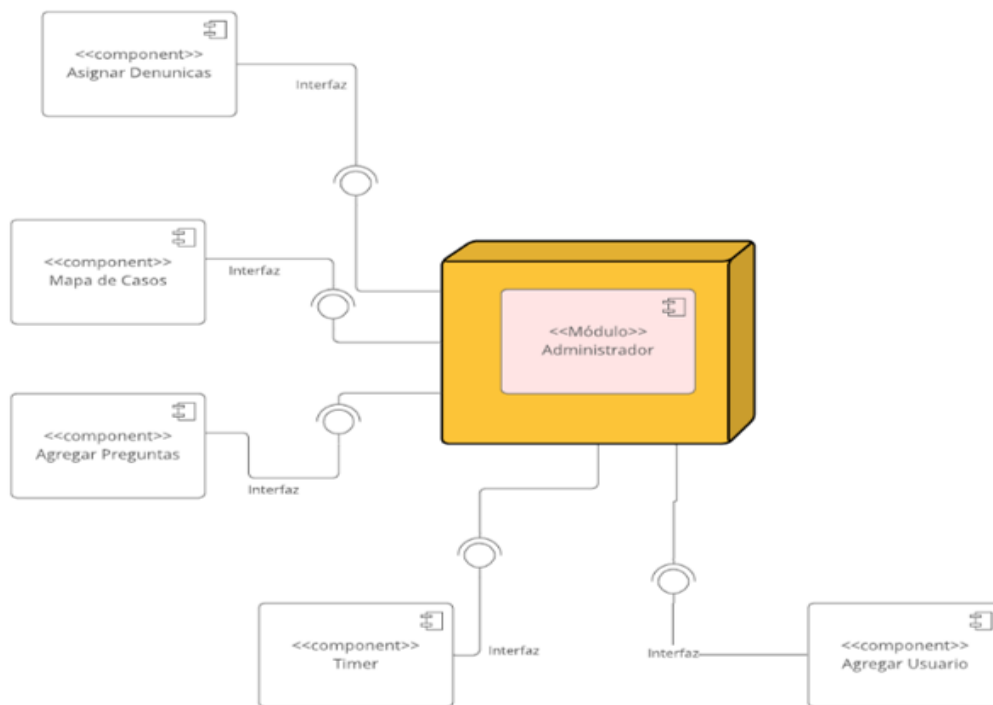


Ilustración 58. Diagrama de componentes Administrador, Fuente Autor

A continuación, se explica la transición entre los diseños mockup hasta llegar a la vista FrontEnd final, dando como respuesta a los requerimientos del sistema: RF009, RF010, RF011, RF012

Vista de pantalla principal administrador

El siguiente mockup (Ilustración 59) muestra la vista principal del módulo Administrador; en esta se sigue el patrón de diseño con el menú a la izquierda, y los campos:

Gestionar denuncia, agregar docente, agregar administrador, mapa de incidencias, cerrar sesión.



Ilustración 59. Vista principal administrador, fuente autor:

Para la transformación del modelo mockup a la vista front final se realizó la codificación en html utilizando el contenedor `` el cual se usa para aplicar estilo al texto o agrupar elementos en línea; también se utilizaron los estilos propios de Bootstrap invitándolos por medio del link `href="css/sb-admin-2.min.css" rel="stylesheet">`.

A continuación, se presenta una parte del proceso de transformación en la estructura front de la vista tomando como ejemplo el componente “asignar denuncia” :

```

<body id="page-top">

<!-- Page Wrapper -->
<div id="wrapper">

<!-- Sidebar -->
<ul class="navbar-nav bg-gradient-danger sidebar sidebar-dark accordion" id="accordionSidebar">

<!-- Sidebar - Brand -->
<a class="sidebar-brand d-flex align-items-center justify-content-center" href="index.html">
  <div class="sidebar-brand-text mx-3">ADMINISTRADOR<br>
  <label id="lblId" style="color:white;"><?php echo($datos_usuario['Id']);?></label>

  </div>
</a>

<!-- Divider -->
<hr class="sidebar-divider">

<!-- Heading -->
<div class="sidebar-heading">
  Denuncias
</div>
<!-- Nav Item - Tables -->
<li class="nav-item">
  <a class="nav-link" href="#" id="btn_asignar_denuncia" onclick="getScreen(1)">
    <i class="fas fa-fw fa-arrow-alt-circle-right"></i>
    <span>Asignar denuncias</span></a>
</li>

```

En la ilustración 60 se puede apreciar la disposición del menú principal para el administrador donde se realizan las siguientes acciones:

Asignación de denuncias, creación de usuarios docentes, creación de usuarios administradores. Visualizar el mapa de denuncias, agregar preguntas al cuestionario y la visualización de la información general.

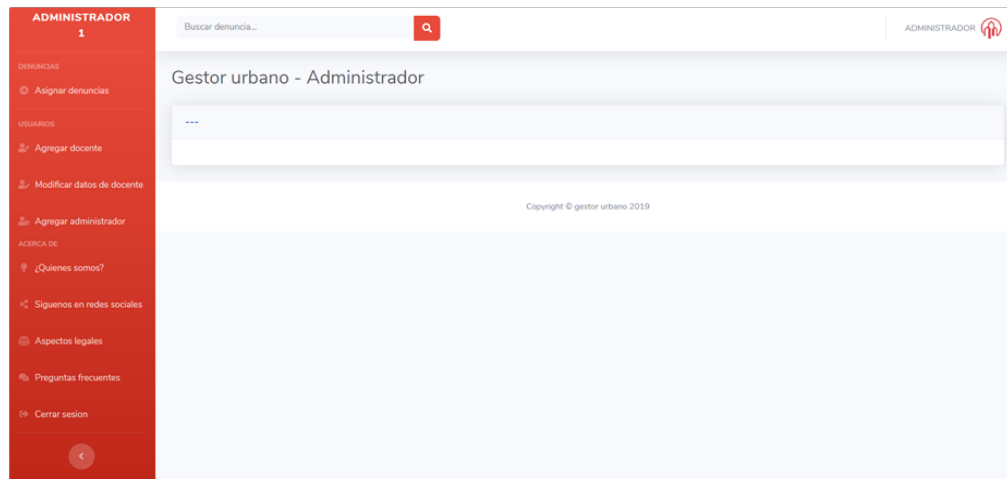


Ilustración 60. Vista frontend página principal-Administrador. Fuente Autor

En la ilustración 60, hubo variación en comparación al diseño mockup de la ilustración 59, ya que se agregaron los siguientes componentes: modificar preguntas del formulario y los modales para la información general.

Para mayor detalle de la codificación de la vista principal del administrador, consultar anexo D, Manual Técnico.

Vista de asignación de denuncias

El diseño de la ilustración 61 corresponde al mockup del componente asignar denuncias; este cuenta con los siguientes campos:

ID denuncia, Estado denuncia descargar reporte, atendida por, y fecha de la denuncia.

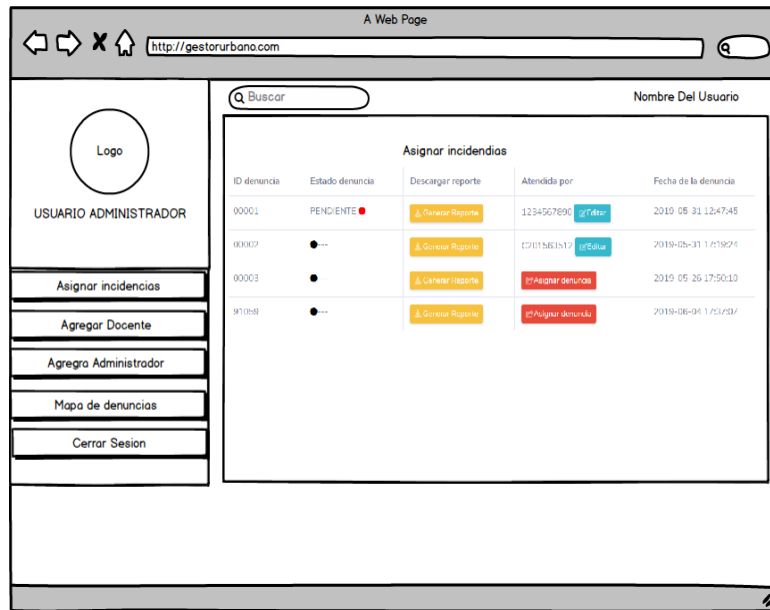


Ilustración 61. Mockup asignación de denuncias, fuente autor

La estructura frontend de la vista 62 está compuesta por un modal asignar denuncia este muestra una nueva vista con los datos de los docentes gracias al método `teachers()`, el estilo de los botones cancelar y reasignar es importado de la librería de Bootstrap.

```

1 <?php
2 // require("../classes/teacher_data.php");
3 // $datos_docentes = new teachers();
4 ?>
5 <div class="modal fade" id="modal-reasignar-denuncia" tabindex="-1" role="dialog" aria-labelledby="exampleModalCenterTitle" aria-hidden="true"
6 <div class="modal-dialog modal-dialog-centered" role="document">
7 <div class="modal-content">
8 <div class="modal-header">
9 <h5 class="modal-title" id="exampleModalCenterTitle">Reasignar denuncia</h5>
10 <button type="button" class="close" data-dismiss="modal" aria-label="Close">
11 <span aria-hidden="true">&times;</span>
12 </button>
13 </div>
14 <div class="modal-body">
15 Esta denuncia actualmente se encuentra asignada al docente con id <span id="Ida"><b><?php echo($idDocente); ?></b></span> y será
16 <select id="select-reasignar-docente">
17 <?php
18 $lista_docentes = $datos_docentes->getAllDocList();
19 echo $lista_docentes;
20 ?>
21 </select>
22 <br><br>
23 <label id="consolea-reasignar-denuncia"></label>
24 </div>
25 <div class="modal-footer">
26 <button type="button" class="btn btn-secondary" data-dismiss="modal">CANCELAR</button>
27 <button type="button" class="btn btn-success" id="btn-reasignar-denuncia">Reasignar</button>
28 </div>

```

Como resultado se obtiene la siguiente vista:

Ilustración 62. Asignar denuncia a docente, fuente autor

Esta consta de un menú de selección donde se listan los docentes creados.

Lo que realiza el sistema internamente es requerir la clase teachers data.php y luego se utiliza la estructura de control “if” para validar los datos del docente.

```

require("../clases/teacher_data.php");
$datos_docentes = new teachers();
if(isset($_POST['nombre_docente']) && isset($_POST['id_denuncia'])){
    $docente = $_POST['nombre_docente'];
    $id_denuncia = $_POST['id_denuncia'];
    $codigo_docente = $datos_docentes->getIdOfDoc($docente);
    $codigo_docente = $codigo_docente['codigoDocente'];
    $asignar = $datos_docentes->updateComplaintAssignedDoc($id_denuncia,$codigo_docente);
    echo $asignar;
}else{
    echo "Consulta inválida";
}

?>

```

Con la ayuda de las herramientas de Bootstrap se logró estilizar y automatizar la parte visual de la asignación de las denuncias; en esta parte del desarrollo se observa como las denuncias constan de un ID generado automáticamente por el sistema, el cual lleva un consecutivo solicitado por el cliente. Además del estado de la denuncia, este estado inicia en pendiente y a medida que la denuncia es gestionada, va variando según el estado en el que se encuentre. Además de esto se puede descargar el reporte o la denuncia, este se imprimirá en pantalla en formato PDF. También la asignación de denuncias a un determinado docente y finalmente la fecha y hora de generación de la denuncia (Ver Ilustración 63).

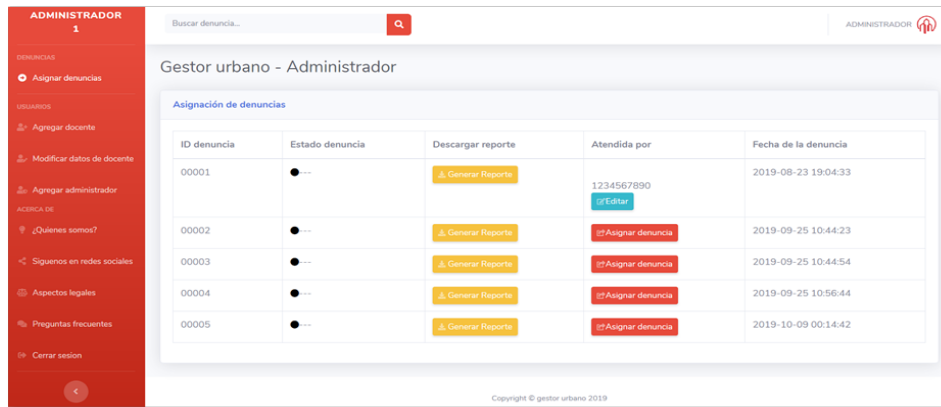


Ilustración 63. Vista frontend asignación de denuncia, fuente autor

Para mayor detalle de la codificación de la asignación de denuncias, anexo D, Manual Técnico.

Agregar preguntas al formulario

Se agregó una funcionalidad a la plataforma como lo muestra la ilustración número 64, la cual permite agregar o quitar de forma dinámica preguntas al formulario. Para ello se utilizó la librería de Bootstrap con la cual se les dio estilo a los botones y a la vista en general.

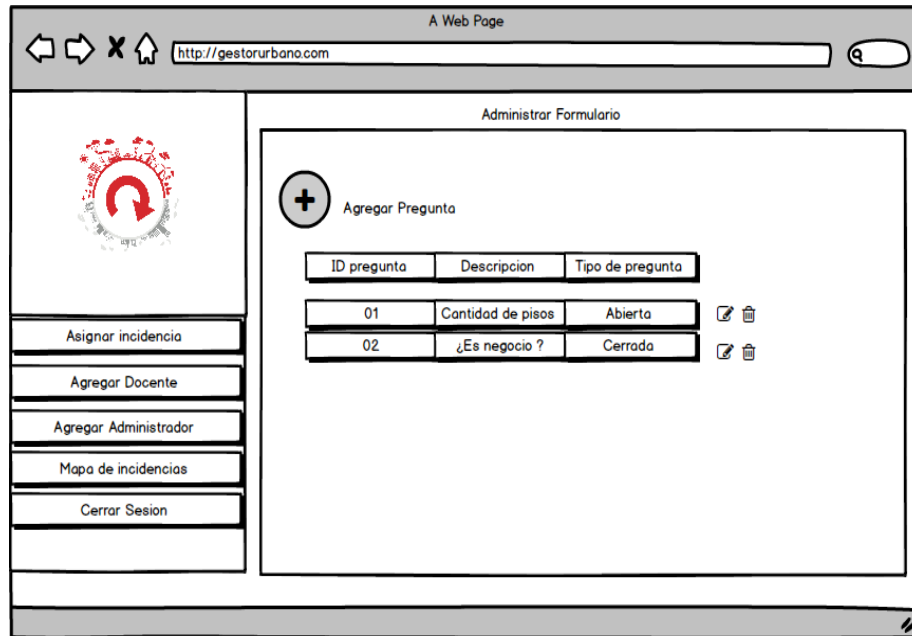


Ilustración 64. Mockup Agregar preguntas al formulario, fuente autor

Al momento de presionar el botón “agregar denuncia” se despliega la vista de la Ilustración 65.

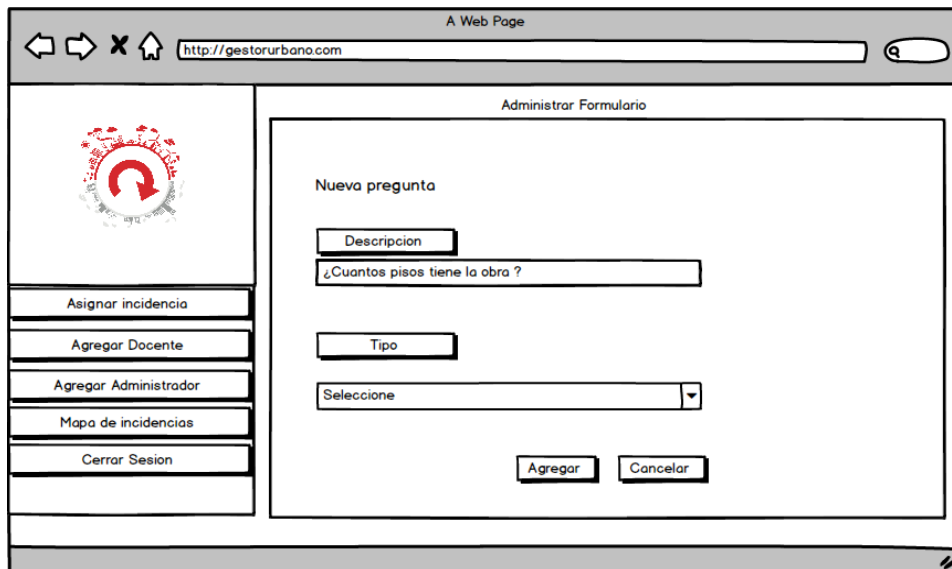


Ilustración 65. Mockup Agregar preguntas al formulario, fuente autor

Para la construcción de la vista se utilizó la siguiente codificación que, básicamente lo que hace es mostrar las preguntas en el cuestionario

```
<?php
$i=1;
$consulta= "SELECT * from preguntas_cuestionario";
$query = mysqli_query($conexion, $consulta);
while ($nt=mysqli_fetch_array($query)){
?>
<tr>
<td>{<?php echo $nt['pregunta'];?>
<i class="fas fa-fw fa-info" onclick="info(<?php echo $i;?>)" style="cursor:pointer;"></i>
</td>
<td>
<input type="radio" id="radioAnswer_<?php echo $i+1;?>_s" name="Answer_<?php echo $i+1;?>" class="custom-control-input" value="SI">
<label class="custom-control-label" for="radioAnswer_<?php echo $i+1;?>_s">SI</label>
</td>
<td>
<input type="radio" id="radioAnswer_<?php echo $i+1;?>_n" name="Answer_<?php echo $i+1;?>" class="custom-control-input" value="NO">
<label class="custom-control-label" for="radioAnswer_<?php echo $i+1;?>_n">NO</label>
</td>
</tr>
<?php
$i=$i+1;
} ?>
```

Luego para la modificar, agregar y eliminar las preguntas se codificó aquí se utiliza un modal traído de la clase data target (propia de bootstrap) la clase table- striped la cual nos muestra las columnas de la tabla, las etiquetas <td> que se utilizó para definir las celdas de la tabla como se muestra en el siguiente fragmento de código:

```

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#modal_agregar_preguntas">
<i class="fas fa-fa fa-plus-circle"></i>
  Agregar Pregunta</button>
<br /> <br />
<table class="table table-striped jambo_table bulk_action" id="tabla_Preguntas">
  <thead class="headings">
    <tr>
      <td>ID PREGUNTA</td>
      <td>PREGUNTA</td>
      <td>TIPO</td>
      <td ></td>
    </tr>
  </thead>
  <tbody class="even pointer" id="tabla_cuerpo">
    <?php
    while ($pregunta=mysqli_fetch_array($query)){
    >
    <tr>
      <td><?php echo $pregunta['id']; >></td>
      <td><?php echo $pregunta['pregunta']; >></td>
      <td><?php echo $pregunta['tipo']; >></td>
      <td>
        <span class="pull-right">
          <a href="#" class="btn btn-light btn-sm" title='Editar pregunta' onclick="listarPorIdPregunta('<?php echo $pregunta['id']; >')"
            data-toggle="modal" data-target="#modal_editar_pregunta"><i class="far fa-edit"></i></a>
          <a href="#" class="btn btn-light btn-sm" title='Eliminar pregunta' onclick="eliminarPregunta('<?php echo $pregunta['id']; >')">
            <i class="fas fa-trash-alt"></i></a>
        </span>
      </td>
    </tr>
    <?php >>
  </tbody>
</table>

```

Como resultado final se obtuvieron las vistas de las Ilustraciones 66 y 67:



Ilustración 66. Vista Frontend Agregar preguntas al cuestionario, Fuente Autor



Ilustración 67. Vista Frontend agregar preguntas al cuestionario, Fuente Autor

Para mayor detalle de la codificación de administración del formulario, consultar anexo D, Manual Técnico.

Timer:

El timer es una funcionalidad del sistema para identificar aquellas denuncias por estado teniendo en cuenta el momento de la creación de esta; de tal modo que el usuario administrador y el usuario docente puedan llevar un control en cuanto a la oportunidad de la respuesta.

Para el diseño de esta funcionalidad se utilizaron las variables `fecha actual` y `fecha_final`, estas, junto con la función `strtotime()` la cual convierte una cadena con formato de fecha en otra con formato de fecha Unix, de tal manera que al tomar los parámetros `d-m-y h:i:s`, los cuales toman la fecha y hora con un condicional definido por la estructura de control "if" para asignarle un color a la denuncia dependiendo los parámetros de inicio y finalización de esta, el siguiente código muestra la estructura del timer:


```

$fecha_actual =strtotime(date("d-m-Y H:i:s"));
$fecha_final_rojo=strtotime(date("d-m-Y H:i:s",strtotime($Mc[$i]['fecha']."+ 4 days")));
$fecha_final_naranja1=strtotime(date("d-m-Y H:i:s",strtotime($Mc[$i]['fecha']."+ 2 days")));
$fecha_final_naranja2=strtotime(date("d-m-Y H:i:s",strtotime($Mc[$i]['fecha']."+ 4 days")));

$tablaD="<tr>
<td>". $Mc[$i]['idDenuncia']. "</td>
<td>". $estado. "</td>
<td>
<a href='../HTML2PDF/denuncia.php?id=" . $Mc[$i]['idDenuncia'] . "' target='_blank' class='d-sm-inline-block btn btn-sm btn-warning shadow-sm
</td>
<td>". $btn_Reasignar. "</td>";

if($fecha_actual >= $fecha_final_rojo){
if($Mc[$i]['estado'] == "FINALIZADA"){
$tablaD=$tablaD."<td style=' color: red; '>". $Mc[$i]['fecha']. "</td></tr>";
}else{
$tablaD=$tablaD."<td style='background:#FC9794; color: red; font-weight: bold; '>". $Mc[$i]['fecha']. "</td></tr>";
}
}else if($fecha_actual > $fecha_final_naranja1 && $fecha_actual < $fecha_final_naranja2){
if($Mc[$i]['estado'] == "FINALIZADA"){
$tablaD=$tablaD."<td style=' color: orange; '>". $Mc[$i]['fecha']. "</td></tr>";
}else{
$tablaD=$tablaD."<td style='background:#FDD58E; color: orange; font-weight: bold; '>". $Mc[$i]['fecha']. "</td></tr>";
}
}else{
if($Mc[$i]['estado'] == "FINALIZADA"){
$tablaD=$tablaD."<td style=' color: green; '>". $Mc[$i]['fecha']. "</td></tr>";
}else{
$tablaD=$tablaD."<td style='background:#A3FB77; color: green; font-weight: bold; '>". $Mc[$i]['fecha']. "</td></tr>";
}
}
}

echo $tablaD;

```

La funcionalidad del Timer consiste en: cada denuncia tiene un máximo de tiempo de respuesta de 5 días hábiles. El sistema empieza a contar desde el momento en que se genera la denuncia y esta queda visible con el color verde. Una vez transcurridos los tres primeros días, la denuncia cambia a color naranja con el fin de alertar al personal encargado. Transcurridos 5 días, la denuncia cambia a color rojo indicando que la denuncia no fue respondida a tiempo y el administrador tomará las respectivas acciones. Si la denuncia es respondida en el plazo está cambiará a color verde. Para más información acerca del timer, consultar el manual de usuario. La vista final quedó de la siguiente manera (ver Ilustración 68):

Gestor urbano - Administrador

Asignación de denuncias

ID denuncia	Estado denuncia	Descargar reporte	Atendida por	Fecha de la denuncia
00001	●---	Generar Reporte	1234567890 Editar	2019-08-23 19:04:33
00002	●---	Generar Reporte	Asignar denuncia	2019-10-13 10:44:23
00003	●---	Generar Reporte	Asignar denuncia	2019-09-25 10:44:54
00004	●---	Generar Reporte	Asignar denuncia	2019-09-13 00:00:00
00005	●---	Generar Reporte	Asignar denuncia	2019-10-13 00:00:00
00006	●---	Generar Reporte	Asignar denuncia	2019-10-16 11:16:11

Ilustración 68. Vista Frontend timer, Fuente Autor

Para mayor detalle de la codificación del timer, consultar anexo D, Manual Técnico.

Agregar usuario

El siguiente mockup ilustra la estructura de la vista agregar usuario, ya sea para agregar un Administrador o un usuario Docente; inicialmente se planteó en el diseño mockup contará con los datos Nombres Dirección y teléfono como se muestra a continuación (ver Ilustración 69):

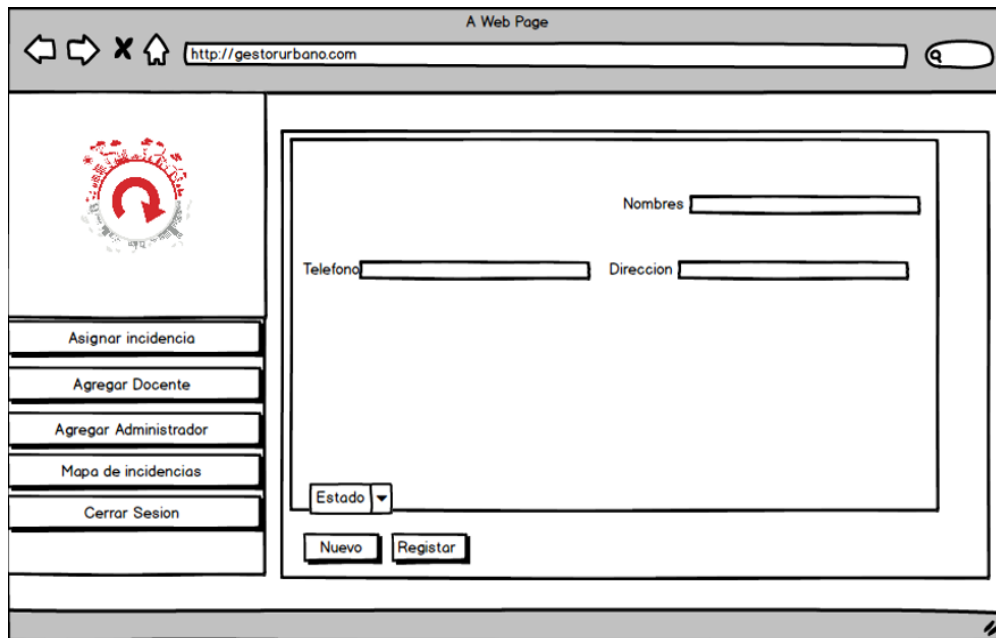


Ilustración 69. Agregar usuario, fuente autor

La vista agregar usuario fue diseñada con las etiquetas HTML `<label>` para los datos de ID, nombres, dirección y demás datos correspondientes a la vista, se utilizaron estilos propios de bootstrap para los campos de texto.

```

6 <div style="display: none;" id="screen2">
7 <label id="idAleatory" style="display: none;"><?php echo("ID del docente ".$id_new_user); ?></label>
8 <form>
9 <div class="form-row">
10 <div class="form-group col-md-6">
11 <label for="txtTeacherName">Nombre completo del docente</label>
12 <input type="text" class="form-control" id="txtNombreT" autocomplete="off">
13 </div>
14 <div class="form-group col-md-6">
15 <label for="txtTeacherAdress">Direccion</label>
16 <input type="text" class="form-control" id="txtDireccionT" autocomplete="off">
17 </div>
18 </div>

```

Se utilizó la función `registerUser` para agregar los usuarios a la base de datos utilizando la consulta `INSERT INTO` con los respectivos parámetros como lo muestra el siguiente fragmento de código:

```

public function registerUser($nombre,$direccion,$cedula,$expedicionCedula,$telefono,$usuario,$email,$clave_hash){
    $id= $this->randomId(5);
    if($this->userRegisterVerify($email,$cedula) < 1){
        $consulta = "INSERT INTO usuario (nombre,direccion,cedula,lugarExpedicion,telefono,nombreUsuario,email,password,idU
        values ('$nombre','$direccion','$cedula','$expedicionCedula','$telefono','$usuario','$email','$clave_hash','$id')";
        // $query=$this->conexion
        $query = $this->conexion->query($consulta);
        if(!$query){
            return "E01";
        }else{
            return true;
        }
    }else{
        return "E00";
    }
}
}

```

La Ilustración 70 muestra la vista final de agregar usuarios, esta acción solo la puede realizar el administrador; este puede agregar docentes y administradores. Se agregó la opción de confirmar la contraseña de acceso al momento de agregar un administrador. También se puede observar que hubo variación en la vista de agregar usuarios, agregando nuevos campos.

The screenshot displays the 'Gestor urbano - Administrador' web application. On the left is a red sidebar menu with options like 'Asignar denuncias', 'Agregar docente', 'Modificar datos de docente', 'Agregar administrador', and 'Cerrar sesion'. The main content area is titled 'Gestor urbano - Administrador' and contains a form titled 'Agregar docente'. The form has the following fields:

- Nombre completo del docente (text input)
- Direccion (text input)
- Telefono (text input)
- Correo electronico (text input)
- Cedula (text input)
- Lugar de expedicion (text input)
- Codigo Docente (text input, containing 'C000000000')
- Contraseña (password input, containing '*****')
- Repetir contraseña (password input, containing '*****')

At the bottom of the form is a green 'Registrar' button with a checkmark icon.

Ilustración 70. Vista Frontend agregar usuario, Fuente Autor

Para mayor detalle de la codificación de agregar usuarios, anexo D, Manual Técnico.

Módulo docente

En la ilustración 71, se muestran los componentes del módulo Docente y las vistas relacionadas al mismo como lo es: el cambio de contraseña, denuncias asignadas y agregar notas al caso.

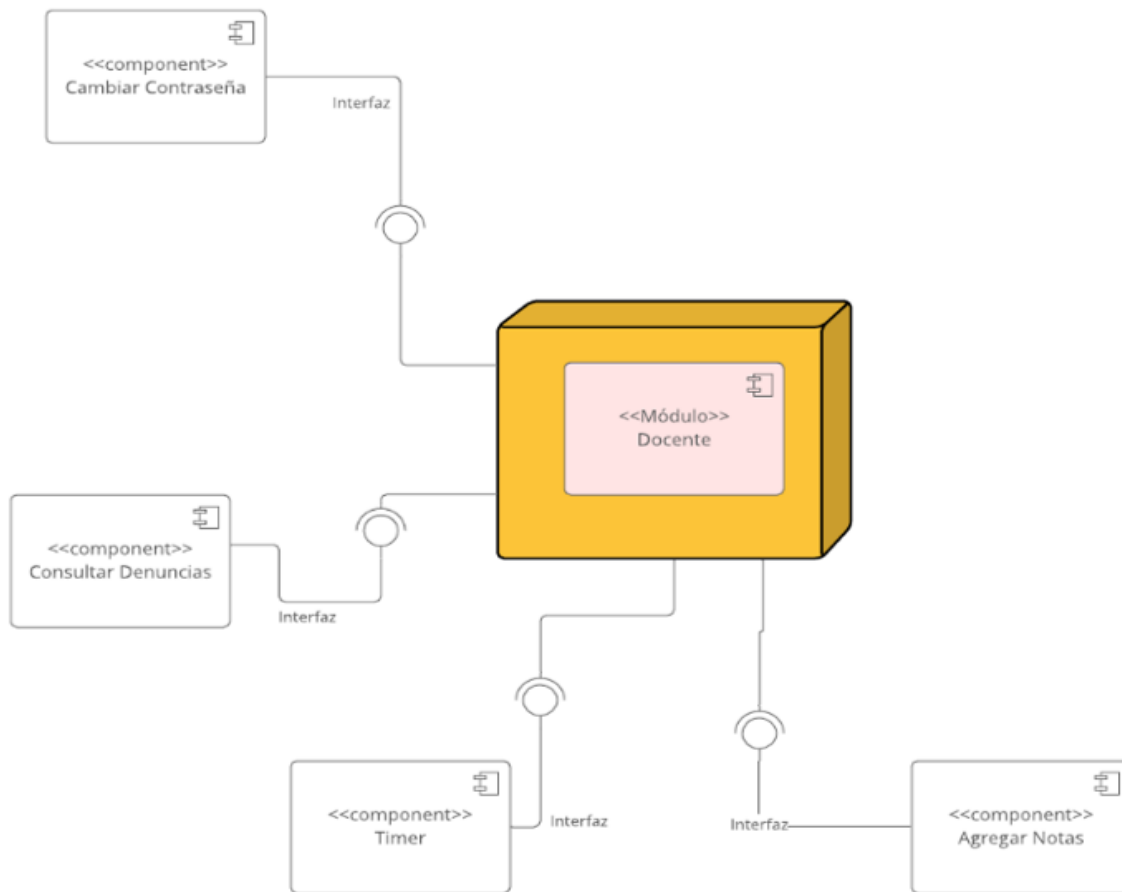


Ilustración 71. Diagrama de componentes, módulo docente, Fuente Autor

A continuación, se explica la transición entre los diseños mockup hasta llegar a la vista FrontEnd final dando como respuesta a los requerimientos del sistema: RF013, RF014:

Actualización de contraseña

El siguiente mockup (Ilustración 72) representa el diseño inicial para la vista "cambio de contraseña", esta utiliza tres campos de textos para diligenciar la contraseña anterior, ingresar contraseña y la confirmación de la misma.

Ilustración 72 muestra un mockup de una interfaz de usuario en un navegador web. El navegador muestra la URL `http://gestorurbano.com` y el título "A Web Page". El encabezado de la página incluye un campo de búsqueda con el texto "Buscar" y el nombre de usuario "Nombre Del Usuario".

El contenido principal está dividido en una barra lateral izquierda y un área de contenido principal. La barra lateral izquierda contiene un círculo con el texto "Logo" y el texto "USUARIO DOCENTE" debajo. Debajo de esto hay un menú con los siguientes ítems: "Lista de casos", "Mapa de casos", "Cambio de contraseña" (destacado), y "Cerrar Sesión".

El área de contenido principal tiene el título "Cambio de contraseña". Contiene tres campos de texto con las etiquetas "Contraseña anterior", "Nueva contraseña" y "Repetir contraseña". Debajo de estos campos hay un botón "Guardar".

Ilustración 72. Cambio de contraseña, fuente autor

Los campos de textos están diseñados con las etiquetas HTML `<input >` y la etiqueta `<label >` para los nombres de los campos con sus respectivo ID

En el siguiente fragmento de códigos se puede observar la construcción de la vista cambio de contraseña:

```
<div id="contenedorFormularioCambioClave" class="text-center shadow p-3 mb-5 bg-white rounded">
  <form class="form-signin">
    <label for="inputClave">Contraseña Antigua</label>
    <input type="password" class="form-control" id="inputClave-antigua" placeholder="*****" pattern="[A-Za-z0-9]">
    <label for="inputClave">Nueva contraseña</label>
    <input type="password" class="form-control" id="inputClave-nueva" placeholder="*****" pattern="[A-Za-z0-9]{4}">
    <label for="inputClave">Repetir nueva contraseña</label>
    <input type="password" class="form-control" id="inputClave-nueva-r" placeholder="*****" pattern="[A-Za-z0-9]">
    <label id="consola"></label><br>
    <button type="submit" id="btn-cambiar-clave" class="btn btn-danger">ACTUALIZAR<i class="fas fa-fw fa-sync">
    <br>
    <label id="label-correo"><?php echo($_SESSION['teacher']['Email']); ?></label>
    <br>
  </form>
</div>
```

Como resultado final se tiene la vista frontend final de la ilustración 73; en esta vista se muestran los campos necesarios para la actualización de contraseña. Esta puede ser cambiada por el usuario docente en cualquier momento. Estos campos son almacenados en base de datos y cuentan con el hash MD5.

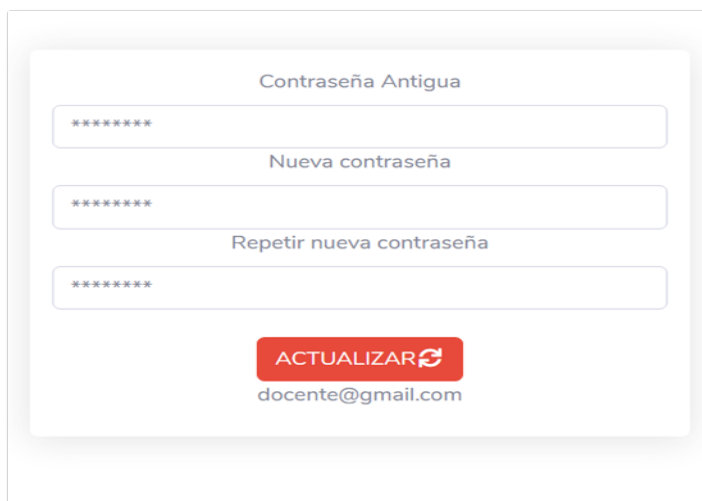


Ilustración 73. Vista Front log in - Fuente autor

Para mayor detalle de la codificación del cambio de contraseña, anexo D, Manual Técnico.

Denuncias asignadas

El siguiente mockup (ilustración 74), muestra el diseño inicial de la vista para consultar las denuncias asignadas a docente donde se muestra el ID de la denuncia, Estado de la denuncia, Descargar reporte, Atendida por y fecha de la denuncia:



Ilustración 74. Mockup lista de casos, Fuente autor

En este módulo del se desarrollaron las vistas para gestión y respuesta de las denuncias asignadas, en el siguiente fragmento de código se puede observar que se agregaron nuevas funcionalidades como agregar notas al caso estos se encuentra dentro de <div>.

Estructura Frontend modificar estados de la denuncia

```
<div class="modal fade" id="alerta-actualizacion-estado-denuncia" tabindex="-1" role="dialog" aria-labelledby="exampleModallabel" aria-hidd
<div class="modal-dialog" role="document">
  <div class="modal-content">
    <div class="modal-header">
      <h5 class="modal-title" id="exampleModallabel">Actualizan de estado denuncia</h5>
      <button type="button" class="close" aria-label="Close" onclick="actualizar()">
        <span aria-hidden="true">&times;</span>
      </button>
    </div>
  </div>
```

Función para mostrar las denuncias asignadas

```
public function getMyAssignedComplaints($idDocente,$filtro = false){
    switch ($filtro) {
        case 'All':
            $consultaDenuncias= "SELECT * from denuncia WHERE idDocente= '".$idDocente.'";
            break;

        default:
            $consultaDenuncias= "SELECT idDenuncia,fecha from denuncia WHERE idUsuario='".$idUsuario.'";
            break;
    }
}
```

Estructura frontend para actualizar estado de denuncia:

```

<h3>Actualizar denuncia <b><span id="id-denuncia"></span></b></h3>
<label for="select-nuevo-estado">Nuevo estado</label>
<SELECT class="form-control" id="select-nuevo-estado">
  <option>SELECCIONE</option>
  <option>EN ESTUDIO</option>
  <option>EN TRAMITE</option>
  <option>PRESENTADA</option>
  <option>FINALIZADA</option>
  <option>RECHAZADA</option>

```

Función para actualizar estado de denuncia

```

1  <?php
2  if(isset($_POST['nuevo_estado']) && isset($_POST['id'])){
3
4      $idDenuncia = $_POST['id'];
5      $nuevo_estado= $_POST['nuevo_estado'];
6
7      require("../unilab_clases/complaint_class.php");
8
9      $Complaint = new Complaint ();
0
1      $result = $Complaint->updateState($idDenuncia,$nuevo_estado);
2
3      echo($result);
4  }

```

La Ilustración 75 muestra la vista final de la opción para agregar notas al caso. Los docentes pueden ir actualizando la información relevante de cada uno de los casos, la cual puede ser consultada en los demás módulos del sistema.

Agregar nota

Detalles de la nota:

Cancelar Guardar

Ilustración 75. Vista Front agregar nota, Fuente autor

La ilustración 76 muestra el diseño final de la vista en esta se agregó un campo llamado “Notas” por lo cual se indica una pequeña variación en cuanto al mockup de la ilustración 74.

DOCENTE 1234565467

Buscar denuncia...

DOCENTE

Gestor urbano - Docente

Gestion de denuncias

ID denuncia	Estado denuncia	Notas	Descargar reporte	Fecha de la denuncia
00002	EN ESTUDIO C	+ -	Generar Reporte	2019-10-11 09:41:29
00003	EN ESTUDIO C	+ -	Generar Reporte	2019-10-11 09:41:41

Copyright © gestor urbano 2019

Ilustración 76. Vista frontend de la gestión de denuncias, Fuente Autor

Para mayor detalle de la codificación de la gestión de denuncias, consultar anexo D, Manual Técnico

5. PRUEBAS DEL SISTEMA

El presente capítulo tiene como objetivo presentar los resultados de las pruebas de interfaz y funcionales del sistema gestor urbano teniendo en cuenta los estándares de calidad aplicando las técnicas de CMMI. Se realizaron las pruebas de funcionales y de interfaz de acuerdo a los requisitos del sistema y teniendo en cuenta el alcance del proyecto, al igual que los modelos conceptuales previamente diseñados, todo esto con el fin de divisar los posibles fallos y posibles errores en el rendimiento del sistema y poder mitigar el número de inconsistencias detectadas, de tal manera, que a la hora de que el software sea utilizado por el usuario final, este sea eficiente y de buena calidad, además de contar con un buen diseño, ser intuitivo y adaptable a las necesidades del usuario.

5.1 Pruebas de interfaz

Las actividades de esta etapa del proyecto se realizaron con el fin de analizar las revisiones precisas de la forma en la que se despliegan las páginas del software y garantizar el cumplimiento de los estándares mínimos definidos en el anexo D, el cual hace referencia al formato utilizado para las respectivas validaciones. A continuación de muestran los resultados obtenidos de la realización de las pruebas de interfaz de los módulos Administrador, Ciudadano y Docente:

Resultado Final

El resultado arrojado en la versión 1.0 de la prueba realizada, fue el más satisfactorio porque estuvo por encima del 99% de aceptabilidad, en la Lista de Chequeos se tienen 23 puntos y 23 ítems de verificación; el nivel de conformidad se encuentra en el 100% de aceptabilidad la siguiente grafica contempla los resultados las pruebas de interfaz:

PRUEBA DE INTERFAZ GRÁFICA PARA CIUDADANO

Escala de calificación:

Conforme	No Conforme	Parcialment e	No aplica
Sección	Calificación	Observación	
1. Interfaz General o Ventana Nueva			
Se visualiza los controles como activos cuando esto es necesario	Conforme		
En la barra de título aparece el nombre representativo de la interfaz	Conforme		
Primera letra en mayúsculas y las demás en minúscula	Conforme		
3. Cuadros de Texto			
¿Los cuadros de texto están alineados correctamente?	Conforme		
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	Conforme		
El ancho es correspondiente con la información que digitará el usuario.	Conforme		
Las cajas de texto ubican al lado derecho de las etiquetas.	Conforme		
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme		
4. Combos			
La longitud del combo corresponde a la información a mostrar	Conforme		
Se ubican al lado derecho de las etiquetas	Conforme		
Los combos están alineados correctamente	Conforme		
5. Botones			
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme		
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme		

Si el botón tiene que abrir una nueva ventana lo hace correctamente	Conforme	
6.Grillas		
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme	
La repartición de las celdas que están dentro de la grilla está bien distribuida.	Conforme	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	
7.Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	Conforme	
Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	Conforme	
¿Se utilizan para alertar al usuario?	Conforme	
Se utilizan para brindar información cuando haya ocurrido un error	Conforme	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	Conforme	
¿Los mensajes desplegados están escritos con buena ortografía?	Conforme	

Tabla 38. Pruebas de interfaz gráfica ciudadano, Fuente autor

5.2 Módulo administrador

Se revisaron las siguientes interfaces con el fin de eliminar el margen de error a nivel visual durante la construcción del front –end:

Tabla 56. Prueba de interfaz gráfica para administrador, fuente autor

PRUEBA DE INTERFAZ GRÁFICA PARA ADMINISTRADOR

Escala de calificación:

Conforme	No Conforme	Parcialmente	No aplica
Sección	Calificación	Observación	
1. Interfaz General o Ventana Nueva			
Se visualiza los controles como activos cuando esto es necesario	Conforme		
En la barra de título aparece el nombre representativo de la interfaz	Conforme		
Primera letra en mayúsculas y las demás en minúscula	Conforme		
3. Cuadros de Texto			
¿Los cuadros de texto están alineados correctamente?	Conforme		
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	Conforme		
El ancho es correspondiente con la información que digitará el usuario.	Conforme		
Las cajas de texto ubican al lado derecho de las etiquetas.	Conforme		
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme		
4. Combos			
La longitud del combo corresponde a la información a mostrar	Conforme		
Se ubican al lado derecho de las etiquetas	Conforme		
Los combos están alineados correctamente	Conforme		
5. Botones			
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme		
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme		
Si el botón tiene que abrir una nueva ventana lo hace correctamente	Conforme		
6. Grillas			
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme		

La repartición de las celdas que están dentro de la grilla está bien distribuida.	Conforme	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	
7.Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	Conforme	
Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	Conforme	
¿Se utilizan para alertar al usuario?	Conforme	
Se utilizan para brindar información cuando haya ocurrido un error	Conforme	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	Conforme	
¿Los mensajes desplegados están escritos con buena ortografía?	Conforme	

Resultado final

El resultado arrojado en la versión 1.0 de la prueba realizada, fue el más satisfactorio porque estuvo por encima del 99% de aceptabilidad, en la Lista de Chequeos se tienen 23 puntos y 23 ítems de verificación; el nivel de conformidad se encuentra en el 100% de aceptabilidad la siguiente grafica contempla los resultados las pruebas de interfaz:

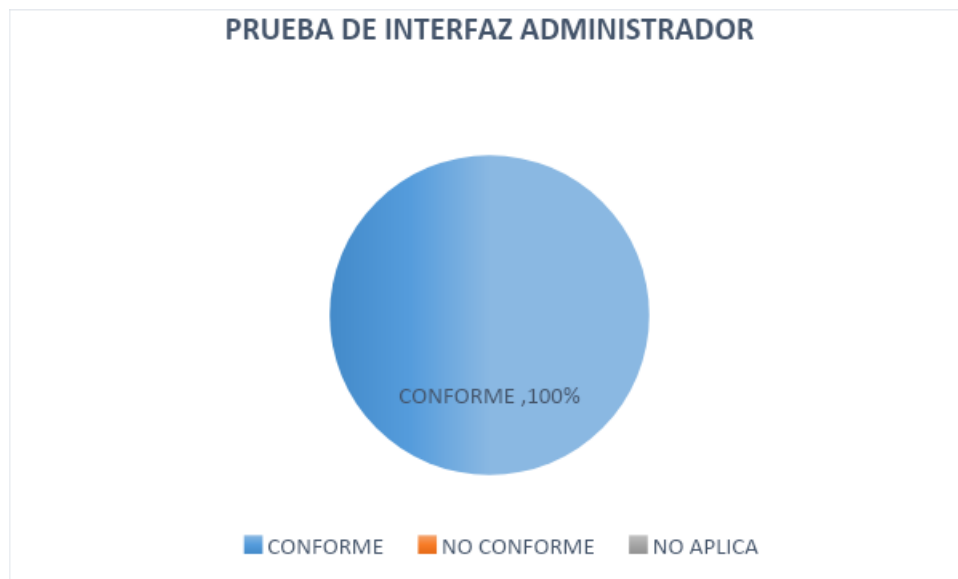


Ilustración 77. Prueba interfaz Administrador, Fuente autor

5.3 Módulo docente

Se revisaron las siguientes interfaces con el fin de eliminar el margen de error a nivel visual durante la construcción del front –end:

Tabla 57. Prueba de interfaz gráfica para docente, fuente autor

PRUEBA DE INTERFAZ GRÁFICA PARA DOCENTE			
<i>Escala de calificación:</i>			
Conforme	No Conforme	Parcialmente	No aplica
Sección	Calificación	Observación	
<i>1. Interfaz General o Ventana Nueva</i>			
Se visualiza los controles como activos cuando esto es necesario	Conforme		
En la barra de título aparece el nombre representativo de la interfaz	Conforme		

Primera letra en mayúsculas y las demás en minúscula	Conforme	
3. Cuadros de Texto		
¿Los cuadros de texto están alineados correctamente?	Conforme	
¿La cantidad de caracteres que permite ingresar, es la cantidad estipulada en el requerimiento o en la tabla de base de datos?	Conforme	
El ancho es correspondiente con la información que digitará el usuario.	Conforme	
Las cajas de texto ubican al lado derecho de las etiquetas.	Conforme	
Entre las etiquetas y los cuadros de texto se deja un espacio prudente	Conforme	
4. Combos		
La longitud del combo corresponde a la información a mostrar	Conforme	
Se ubican al lado derecho de las etiquetas	Conforme	
Los combos están alineados correctamente	Conforme	
5. Botones		
El nombre del botón primera letra en mayúscula y las demás en minúsculas	Conforme	
Si tiene imagen este debe ir al lado izquierdo del texto	Conforme	
Si el botón tiene que abrir una nueva ventana lo hace correctamente	Conforme	
6. Grillas		
El tamaño de las grillas es el adecuado para la información que se desea mostrar	Conforme	
La repartición de las celdas que están dentro de la grilla está bien distribuida.	Conforme	
Los botones que se encuentran dentro de las grillas cumplen con la función necesaria.	Conforme	
7. Mensajes de Información		
¿Los mensajes brindan información conforme a lo requerido?	Conforme	

Se utiliza el icono correspondiente al momento de desplegar los mensajes de información	Conforme	
¿Se utilizan para alertar al usuario?	Conforme	
Se utilizan para brindar información cuando haya ocurrido un error	Conforme	
¿Al momento de brindar información cuando haya ocurrido un error, el mensaje mostrado es claro, explica dónde y porque se está generando el error o como arreglarlo?	Conforme	
¿Los mensajes desplegados están escritos con buena ortografía?	Conforme	

PRUEBA DE INTERFAZ MODULO DOCENTE

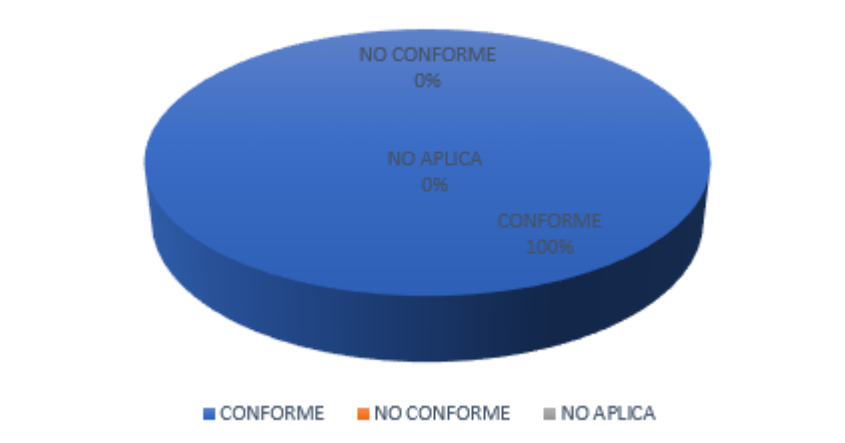


Ilustración 78. Prueba de interfaz de usuario, Fuente autor

Resultado final

El resultado arrojado en la versión 1.0 de la prueba realizada fue el más satisfactorio porque estuvo por encima del 90% de aceptabilidad, en la Lista de Chequeos se tienen 23 puntos y 23 ítems de verificación; el nivel de conformidad se encuentra en el 100% de aceptabilidad la siguiente grafica contempla los resultados las pruebas de interfaz:



Ilustración 79. Prueba de interfaz Administrador, Fuente autor

5.4 Pruebas Funcionales

Las actividades de esta etapa del proyecto se realizaron con el fin de analizar las revisiones precisas de la forma en la que se despliegan las páginas del software y garantizar el cumplimiento de los estándares mínimos de acuerdo con los formatos de severidad y formatos de pruebas funcionales definidos en el anexo 7, el cual hace referencia a los formatos utilizado para las respectivas validaciones. A continuación, se muestran los resultados obtenidos de la realización de las pruebas de interfaz de los módulos Administrador, Ciudadano y Docente:

Ciclo 1.0, Pruebas funcionales con relación a los módulos, Ciudadanos, Administrador y Docente

Se sometieron a pruebas funcionales los siguientes casos de usos que representan requerimientos y una opción dentro del menú de navegación por cada módulo.

Módulo ciudadano

Módulo Ciudadano

- RF-001 Ingresar Ciudadano
- RF-002 Creación De La Denuncia
- RF-003 Modificar Usuario (Actualizar, Editar, Cambiar Contraseña)
- RF-004 Botones De Navegación (Mis Denuncias, Mapa De Casos)
- RF-005 Gestión Y Realización De Formato.
- RF-006 Visualizaciones En El Mapa De Casos.
- RF-007 Adjunta Foto De Evidencia Y Soporte Al Formato
- RF-008 Generar Mensajes De Verificación

Tabla 39. Informe resultado de prueba funcional V 1.0 Modulo Ciudadano Elaborado por los autores

Datos Generales de la Prueba del Sistema	
Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none">● RF-001 Inicio de sesión: usuario● RF-002 creación de la denuncia● RF-003 Permite la actualización de los datos personales de un usuario	Brayan Bravo Silva

<ul style="list-style-type: none"> ● RF-004 Botones de navegación (mis denuncias, mapa de casos) ● RF-005 Llenar los campos para diligenciar el formato (control numérico, check list, agregar dirección, geolocalización y fotos). <ul style="list-style-type: none"> ● RF-006 Visualizaciones en el mapa de casos. ● RF-007 Adjunta foto de evidencia y soporte al formato <ul style="list-style-type: none"> ● RF-008 Generar mensajes de verificación 	
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo para probar:	Módulo Ciudadano
Resultado de las Pruebas del Sistema	
Hallazgo #1	- RF-003 El sistema no adjunta foto de soporte
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote

Fecha: 20 de septiembre de 2019	Fecha: 20 de septiembre de 2019
Firma (Obligatoria):	Firma (Obligatoria):
Recibe el Cliente	
Nombre:	
Fecha: 20 de septiembre de 2019	
Firma (Obligatoria):	

El informe formal presentado al cliente con el resultado de las pruebas, donde se evidencia un total de dos hallazgos que indican errores o fallas en el sistema al momento de realizar una acción.

Módulo administrador

- RF-009 Asignar incidencias
- RF-010 Listar incidencias
- RF-011 Registrar Usuario Administrador
- RF-012 modificar cuestionario

Tabla 40. Informe resultado de prueba funcional V 1.0 Modulo Administrador Elaborado por los autores

Datos Generales de la Prueba del Sistema

Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none"> ● RF-009 Inicio de sesión: Administrador ● RF-010 Crear usuario: Docente ● RF-011 Agregar (docente, administrador), Editar docente, actualizar, Agregar las denuncias a docentes. ● RF-012 Permitir al usuario (administrador) hacer modificaciones a las preguntas el cuestionario para la denuncia. 	<p style="text-align: center;">Brayan Bravo Silva</p>
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Administrador
Resultado de las Pruebas del Sistema	
Hallazgo #1	<ul style="list-style-type: none"> - RF-011 No se actualiza los datos del administrador ni los de los docentes. No se refleja los datos adjuntos de la denuncia.

Hallazgo #1	RF 012 no se implementó según el requerimiento funcional
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 20 de septiembre de 2019	Fecha: 20 de septiembre de 2019
Firma (Obligatoria): <hr/>	Firma (Obligatoria): <hr/>
Recibe el Cliente	
Nombre:	
Fecha: 20 de septiembre de 2019	
Firma (Obligatoria): <hr/>	

El informe formal presentado al cliente con el resultado de las pruebas, donde se evidencia un total de dos hallazgos que indican errores o fallas en el sistema al momento de realizar una acción.

Módulo docente

- RF-013 Inicio de sesión
- RF-014 Listar denuncias asignadas
- RF-015 Asignación de denuncia
- RF-016 Estados de la denuncia
- RF-017 Soportes anexos (notas)
- RF-018 verificación de documento (PDF)
- RF-019 Mensajes de alerta
-

Tabla 41. Informe resultado de prueba funcional V 2.0 Modulo Docente Elaborado por los autores

Datos Generales de la Prueba del Sistema	
Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none">● RF-013 Inicio de sesión: Docente● RF-014 Tabla de denuncias asignadas.● RF-015 escogencia de la denuncia asignada● RF-016 Agregar estados de las denuncias.● RF-017 mensajes de soportes (Notas, imágenes)● RF-018 extracción de documento PDF● RF-019 Generar mensajes de verificación	Brayan Bravo Silva
Descripción de la Prueba del Sistema	

Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Docente
Resultado de las Pruebas del Sistema	
Hallazgo #1	RF-016 no se creó el requerimiento de estados de la denuncia. RF-017 no se visualiza las fotos adjuntas de la denuncia
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 20 de septiembre de 2019	Fecha: 20 de septiembre de 2019
Firma (Obligatoria): <hr/>	Firma (Obligatoria): <hr/>
Recibe el Cliente	
Nombre:	
Fecha: 20 de septiembre de 2019	
Firma (Obligatoria):	

**CICLO 1.0 DE PRUEBAS FUNCIONALES CON RELACIÓN A LOS
MÓDULOS: CIUDADANO, ADMINISTRADOR Y DOCENTE .**

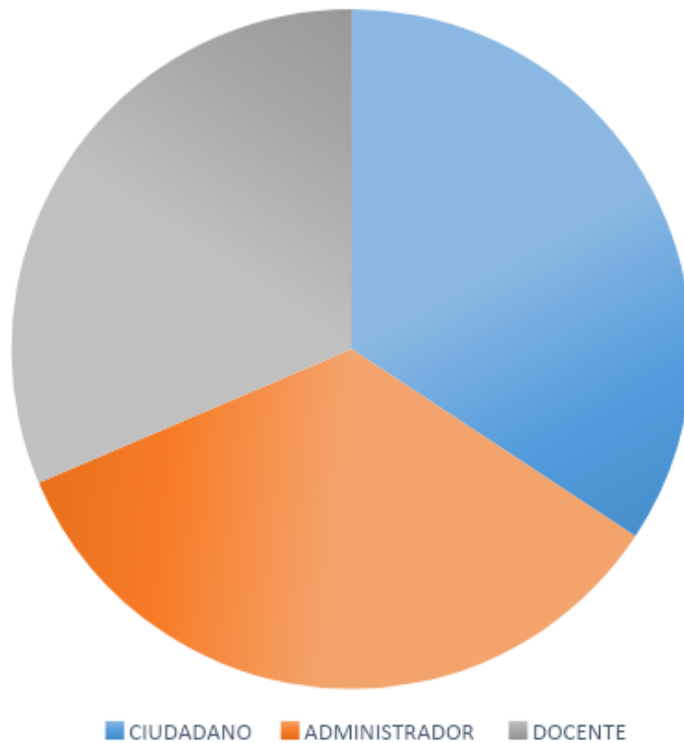


Ilustración 80. Resultados de análisis de pruebas de funcionalidad, Fuente autor

La ilustración 83, muestra el gráfico de los resultados obtenidos de los análisis pruebas de funcionalidad, las cuales se encuentran clasificadas en tres módulos de pruebas tenidos en cuenta en la aplicación.

El primer módulo analizado es el módulo Ciudadano; el cual evidencia fallas para realizar el CRUD correctamente, el adjuntado de las fotos y la geo localización arrojando un resultado de 60 % de funcionalidad; el segundo módulo que es el administrador arrojó un resultado de 60% de funcionalidad; evidenciando el mismo margen de error presentados en los módulos de ciudadano. Para finalizar se analizaron los niveles de funcionalidad del módulo docente, el cual arrojó un resultado de 55% de funcionalidad lo que evidencia fallas en no mostrar mensaje de advertencia al ingresar datos incorrectos (tipo de datos), no visualiza las imágenes adjuntas por el ciudadano.

Con lo anteriormente mencionado, según los resultados obtenidos en las pruebas de funcionalidad llevadas a cabo en cada uno de los módulos pertenecientes a la aplicación GESTOR URBANO se puede concluir que la aplicación obtiene un 65% total de funcionalidad en general de la misma.

Ciclo 2.0, pruebas funcionales con relación a los módulos: Ciudadano, Administrador y docente

En esta etapa iniciamos con esta versión 2 donde evidenciamos las mejoras de la aplicación obteniendo más funcionalidades y satisfacción de la misma.

A continuación, se mostrará las pruebas, si las fallas en el sistema han sido corregidas y evidenciar la disminución de errores.

Modelo ciudadano

- Ingresar Ciudadano
- RF-002 Creación De La Denuncia
- RF-003 Modificar Usuario (Actualizar, Editar, Cambiar Contraseña)
- RF-004 Botones De Navegación (Mis Denuncias, Mapa De Casos)
- RF-005 Gestión Y Realización De Formato.
- RF-006 Visualizaciones En El Mapa De Casos.
- RF-007 Adjunta Foto De Evidencia Y Soporte Al Formato
- RF-008 Generar Mensajes De Verificación

Tabla 42. Informe resultado de prueba funcional V 2.0 Modulo Ciudadano Elaborado por los autores, Fuente autor

Datos Generales de la Prueba del Sistema

Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none"> ● RF-001 Inicio de sesión: usuario ● RF-002 creación de la denuncia ● RF-003 Permite la actualización de los datos personales de un usuario ● RF-004 Botones de navegación (mis denuncias, mapa de casos) ● RF-005 Llenar los campos para diligenciar el formato (control numérico, check list, agregar dirección, geolocalización y fotos). ● RF-006 Visualizaciones en el mapa de casos. ● RF-007 Adjunta foto de evidencia y soporte al formato ● RF-008 Generar mensajes de verificación 	Brayan Bravo Silva
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Ciudadano
Resultado de las Pruebas del Sistema	
Hallazgo	<p>No se encontraron hallazgos e inconformidades en cuanto a la implementación de requerimientos para este módulo, la prueba funcional genero resultados satisfactorios.</p>

Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 04 de octubre de 2019	Fecha: 04 de octubre de 2019
Firma (Obligatoria): _____	Firma (Obligatoria): _____
Recibe el Cliente	
Nombre:	
Fecha: 04 de octubre de 2019	
Firma (Obligatoria):	

El informe formal presentado al cliente con el resultado de las pruebas, donde se evidencia un total de dos hallazgos que indican errores o fallas en el sistema al momento de realizar una acción.

Módulo administrador

- RF-009 Asignar incidencias
- RF-010 Listar incidencias

- RF-011 Registrar Usuario Administrador
- RF-012 modificar cuestionario

Tabla 43. Informe resultado de prueba funcional V 2.0 Modulo Administrador, fuente autor

Datos Generales de la Prueba del Sistema	
Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none"> ● RF-009 Inicio de sesión: Administrador ● RF-010 Crear usuario: Docente ● RF-011 Agregar (docente, administrador), Editar docente, actualizar, Agregar las denuncias a docentes. ● RF-012 Permitir al usuario (administrador) hacer modificaciones a las preguntas el cuestionario para la denuncia. 	Brayan Bravo Silva
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Administrador
Resultado de las Pruebas del Sistema	
Hallazgo #1	- RF-011 No se actualiza los datos del administrador ni los de los docentes
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 04 de octubre de 2019	Fecha: 04 de octubre de 2019
Firma (Obligatoria):	Firma (Obligatoria):
_____	_____

Recibe el Cliente
Nombre:
Fecha: 04 de octubre de 2019
Firma (Obligatoria):

El informe formal presentado al cliente con el resultado de las pruebas, donde se evidencia un total de dos hallazgos que indican errores o fallas en el sistema al momento de realizar una acción.

Modulo docente

- RF-013 Inicio de sesión
- RF-014 Listar denuncias asignadas
- RF-015 Asignación de denuncia
- RF-016 Estados de la denuncia
- RF-017 Soportes anexos (notas)
- RF-018 verificación de documento (PDF)
- RF-019 Mensajes de alerta

Tabla 44. Informe resultado de prueba funcional V 2.0 Modulo Docente, fuente autor

Datos Generales de la Prueba del Sistema	
Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none"> ● RF-013 Inicio de sesión: Docente ● RF-014 Tabla de denuncias asignadas. 	Brayan Bravo Silva

<ul style="list-style-type: none"> ● RF-015 escogencia de la denuncia asignada ● RF-016 Agregar estados de las denuncias. ● RF-017 mensajes de soportes (Notas, imágenes) ● RF-018 extracción de documento PDF ● RF-019 Generar mensajes de verificación 	
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Docente
Resultado de las Pruebas del Sistema	
Hallazgo #1	RF-017 no se visualiza las fotos adjuntas de la denuncia
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 04 de octubre de 2019	Fecha: 04 de octubre de 2019
Firma (Obligatoria): <hr/>	Firma (Obligatoria): <hr/>
Recibe el Cliente	
Nombre:	
Fecha: 04 de octubre de 2019	

Firma (Obligatoria):

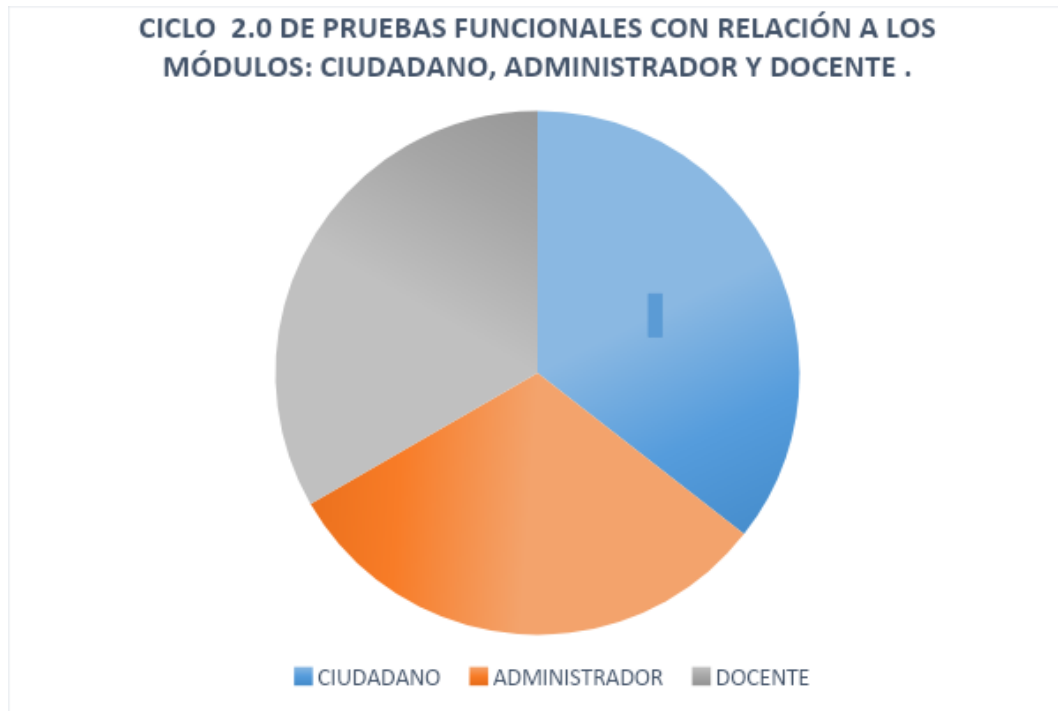


Ilustración 81. Gráfica de resultados, pruebas funciones, clico 2, fuente autor

La ilustración 84, muestra la gráfica donde se evidencia el resultado general de la versión 2 obtenido de las pruebas de funcionalidad llevadas.

El primer módulo analizado es el módulo Ciudadano; el cual se obtiene como resultado 90% de efectividad en su funcionalidad; lo que quiere decir que los resultados arrojados son satisfactorios. El módulo administrador, es el segundo módulo analizado y arroja un 70% de funcionalidad, este módulo evidencia una falla para realizar el CURD correctamente, actualizaciones del docente y la creación del cuestionario. Para finalizar se evaluó el módulo docente, el cual arrojó un resultado de 75% de funcionalidad lo que

evidencia nuevamente fallas en el sistema ya que no visualiza los archivos adjuntos del ciudadano.

Con lo anteriormente planteado, según los resultados obtenidos en las pruebas de funcionalidad llevadas a cabo en cada uno de los módulos pertenecientes a la aplicación GESTOR URBANO se puede concluir que la aplicación obtiene un 76% total de funcionalidad en general de la misma.

Ciclo 3.0, pruebas funcionales con relación a los módulos Ciudadano, Administrador y Docente

El propósito de esta versión de las pruebas es observar si las fallas en el sistema han sido corregidas y evidenciar la disminución de errores.

Módulo ciudadano

- RF-001 Ingresar Ciudadano
- RF-002 Creación De La Denuncia
- RF-003 Modificar Usuario (Actualizar, Editar, Cambiar Contraseña)
- RF-004 Botones De Navegación (Mis Denuncias, Mapa De Casos)
- RF-005 Gestión Y Realización De Formato.
- RF-006 Visualizaciones En El Mapa De Casos.
- RF-007 Adjunta Foto De Evidencia Y Soporte Al Formato
- RF-008 Generar Mensajes De Verificación

Tabla 45. Informe resultado de prueba funcional V 3.0 Modulo Usuario Elaborado por los autores

Datos Generales de la Prueba del Sistema

Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none"> ● RF-001 Inicio de sesión: usuario ● RF-002 creación de la denuncia ● RF-003 Permite la actualización de los datos personales de un usuario ● RF-004 Botones de navegación (mis denuncias, mapa de casos) ● RF-005 Llenar los campos para diligenciar el formato (control numérico, check list, agregar dirección, geolocalización y fotos). ● RF-006 Visualizaciones en el mapa de casos. ● RF-007 Adjunta foto de evidencia y soporte al formato ● RF-008 Generar mensajes de verificación 	<p>Brayan Bravo Silva</p>
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Usuario
Resultado de las Pruebas del Sistema	

Hallazgo	No se encontraron hallazgos e inconformidades en cuanto a la implementación de requerimientos para este módulo, la prueba funcional genero resultados satisfactorios.
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 18 de octubre de 2019	Fecha: 18 de octubre de 2019
Firma (Obligatoria): _____	Firma (Obligatoria): _____
Recibe el Cliente	
Nombre:	
Fecha: 18 de octubre de 2019	
Firma (Obligatoria):	

El informe formal presentado al cliente con el resultado de las pruebas, donde se evidencia un total de dos hallazgos que indican errores o fallas en el sistema al momento de realizar una acción.

Módulo Administrador

- RF-009 Asignar incidencias
- RF-010 Listar incidencias
- RF-011 Registrar Usuario Administrador
- RF-012 modificar cuestionario

Tabla 46. Informe resultado de prueba funcional V 3.0 Modulo Administrador Elaborado por los autores

Datos Generales de la Prueba del Sistema	
Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none">● RF-009 Inicio de sesión: Administrador● RF-010 Crear usuario: Docente● RF-011 Agregar (docente, administrador), Editar docente, actualizar, Agregar las denuncias a docentes.● RF-012 Permitir al usuario (administrador) hacer modificaciones a las preguntas el cuestionario para la denuncia.	Brayan Bravo Silva
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales

Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Administrador
Resultado de las Pruebas del Sistema	
Hallazgo	No se encontraron hallazgos e inconformidades en cuanto a la implementación de requerimientos para este módulo, la prueba funcional genero resultados satisfactorios.
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 18 de octubre de 2019	Fecha: 18 de octubre de 2019
Firma (Obligatoria): _____	Firma (Obligatoria): _____
Recibe el Cliente	
Nombre:	

Fecha: 18 de octubre de 2019

Firma (Obligatoria):

El informe formal presentado al cliente con el resultado de las pruebas, donde se evidencia un total de dos hallazgos que indican errores o fallas en el sistema al momento de realizar una acción.

Módulo Docente

- RF-013 Inicio de sesión
- RF-014 Listar denuncias asignadas
- RF-015 Asignación de denuncia
- RF-016 Estados de la denuncia
- RF-017 Soportes anexos (notas)
- RF-018 verificación de documento (PDF)
- RF-019 Mensajes de alerta

Tabla 46. Informe resultado de prueba funcional V 3.0 Modulo Docente Elaborado por los autores

Datos Generales de la Prueba del Sistema

Características a probar:	Responsable de la Implementación:
<ul style="list-style-type: none"> ● RF-013 Inicio de sesión: Docente ● RF-014 Tabla de denuncias asignadas. ● RF-015 escogencia de la denuncia asignada ● RF-016 Agregar estados de las denuncias. ● RF-017 mensajes de soportes (Notas, imágenes) ● RF-018 extracción de documento PDF ● RF-019 Generar mensajes de verificación 	<p>Brayan Bravo Silva</p>
Descripción de la Prueba del Sistema	
Tipo de prueba:	Pruebas Funcionales
Encargado de pruebas:	Brayan Bravo Silva
Módulo a probar:	Módulo Docente
Resultado de las Pruebas del Sistema	

Hallazgo	No se encontraron hallazgos e inconformidades en cuanto a la implementación de requerimientos para este módulo, la prueba funcional genero resultados satisfactorios.
Validación del Informe	
Encargada de Pruebas	Líder de Proyecto
Nombre: Brayan Bravo Silva	Nombre: Deimer Avila Argote
Fecha: 18 de octubre de 2019	Fecha: 18 de octubre de 2019
Firma (Obligatoria): _____	Firma (Obligatoria): _____
Recibe el Cliente	
Nombre:	
Fecha: 18 de octubre de 2019	
Firma (Obligatoria):	

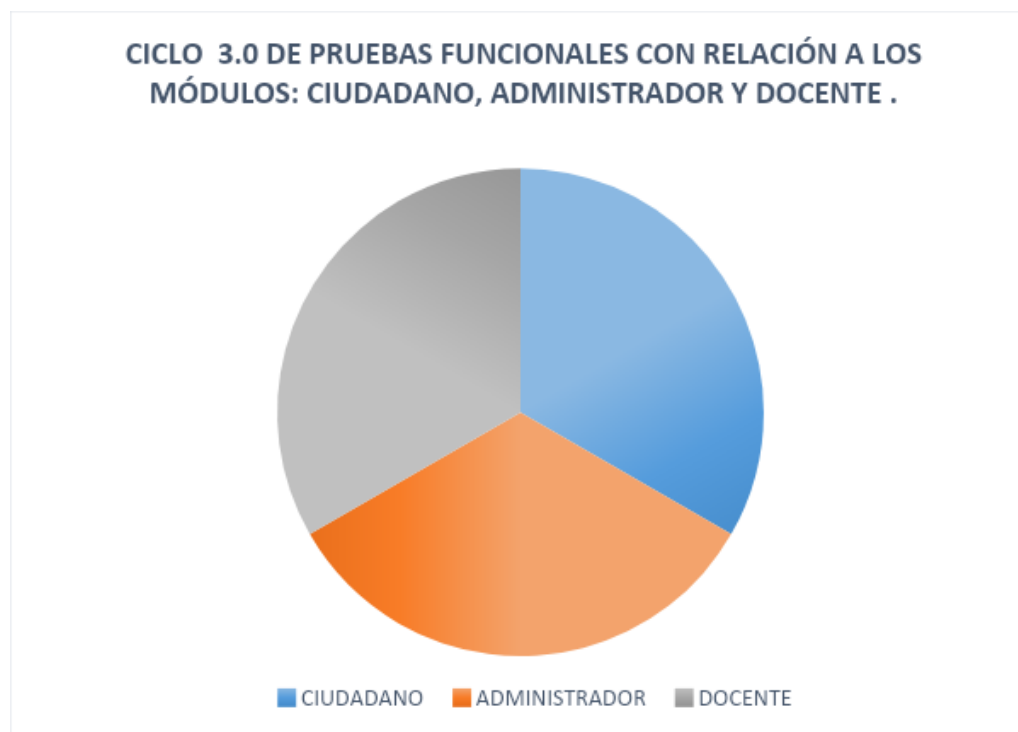


Ilustración 82. Pruebas funcionales ciclo 3. Fuente autor

Finalmente, en la ilustración 85 se muestra la gráfica, donde se evidencia el resultado general de la versión 3 obtenido de las pruebas de funcionalidad.

Tenido en cuenta todos los hallazgos y errores de la aplicación en sus pruebas funcionales, registrados en las 2 versiones anteriores; podemos concluir a cabalidad que los módulos Ciudadano, Administrador y Docente, obteniendo un porcentaje de funcionalidad para cada uno de ellos al 90% se puede concluir que la aplicación está en óptimas condiciones para su operatividad y entrega al cliente.

5.5 Pruebas de severidad

Como complemento de las pruebas funcionales se realizaron las pruebas de severidad para definir por prioridad tomando como criterio de medición los datos severidad baja,

media y alta por cada uno de los módulos (Ciudadano, Docente y administrador) y por cada uno de los ciclos. A continuación, se describen las tablas de datos donde se muestran a detalle los errores por prioridad encontrados en la aplicación según su requerimiento:

Ciclo 1.0

Módulo Administrador

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo Administrador, Ciclo 1.

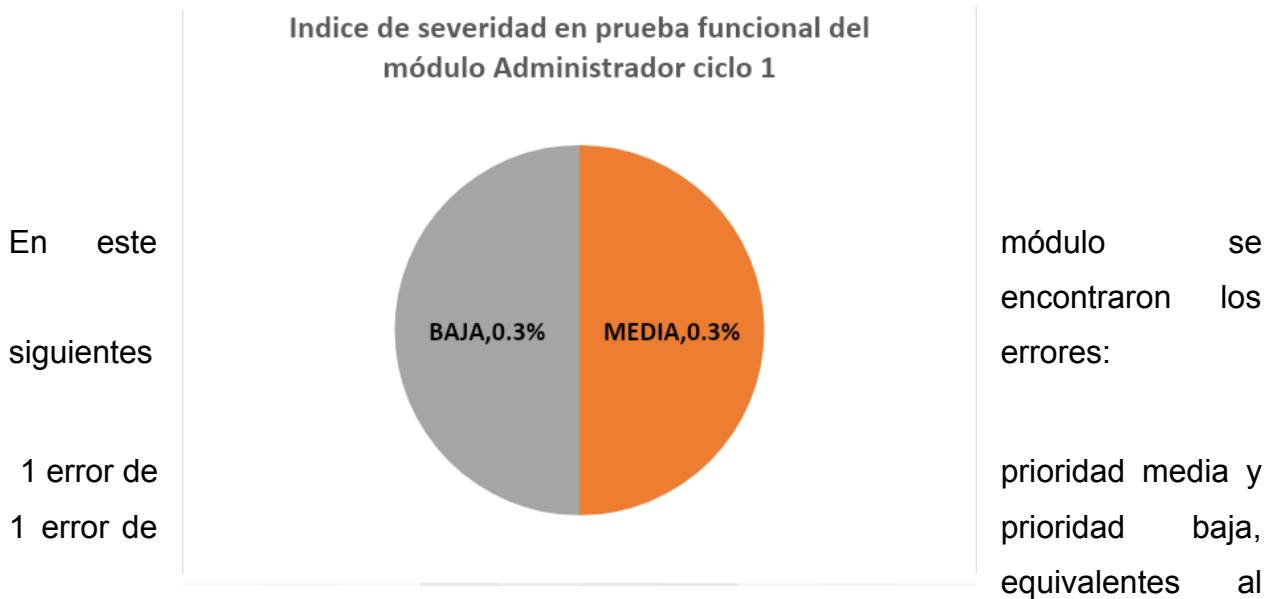
Tabla 47. Pruebas de severidad, ciclo 1, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	Correo electrónico, contraseña, tipo de usuario	Se procede a llenar los campos requeridos para ingresar como usuario admin	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	Alta	Nombre, correo electrónico,	Se procede a llenar los campos	Mensaje de proceso exitoso.	NO

		dirección, cedula, lugar de expedición, código de docente, contraseña	requeridos para crear un nuevo docente y hacer el registro en la base de datos		
Pruebas Funcionales	Media	Nombre, correo electrónico, dirección, cedula, lugar de expedición, código de docente, contraseña	Muestra en pantalla un modal en donde se actualizan los datos del docente,	Mensaje de proceso exitoso	SI
Pruebas Funcionales	Alta	Asignar docente botón (editar)	Se visualizará un botón este asignará las denuncias al docente	No hay Mensaje	NO
Pruebas Funcionales	Baja	Datos del Administrador	Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia	No hay Mensaje	SI

			explicando que ha ocurrido un error		
Pruebas Funcionales	Baja	Datos Entrada	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje de proceso exitoso.	NO

La siguiente grafica hace referencia a los porcentajes de error de acuerdo con la severidad en el ciclo 1 en el módulo administrador:



40% por ciento de errores en el módulo. Estos errores se presentaron en la actualización de registros de los docentes, la validación de los caracteres mínimos se necesita para la identificación del código de los docentes, así mismo los errores según el tipo de datos cada registro.

Módulo ciudadano

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo Ciudadano, Ciclo 1:

Tabla 48. Pruebas de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	Correo electrónico, contraseña, tipo de usuario	Se procede a llenar los campos requeridos para ingresar como usuario	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	Alta	Dirección, Agregar ubicación, Agregar imágenes	Se procede a llenar los campos requeridos para crear una nueva	Mensaje de confirmación si se realizó la acción.	NO

			denuncia hacer el registro en la base de datos		
Pruebas Funcionales	Alta	Check list, control numérico	se diligencia el formulario para generar la denuncia	asignación de datos	NO
Pruebas Funcionales	Alta	Botón mis denuncias, botón mapa de casos, botón de nueva denuncia	Muestra en pantalla todas las funcionalidades donde se realiza la denuncia	información para diligenciar	NO
Pruebas Funcionales	Alta	Botón actualizar mis datos, Botón cambiar contraseña	Se visualizará los botones de entrada, en donde se actualice los datos y contraseña del usuario.	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	media	Mapa de casos	Se visualiza en el mapa todas las denuncias realizadas por	Denuncias agregadas en el mapa	SI

			todos los usuarios.		
Pruebas Funcionales	Alta	imágenes	Se adjunta foto de evidencia y soporte al formato	Imágenes de la denuncia	SI
Pruebas Funcionales	Baja	Botón enviar respuesta	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje de confirmación si se realizó la acción.	NO

La siguiente grafica hace referencia a los porcentajes de error de acuerdo con la severidad en el ciclo 1 en el módulo Ciudadano:



Ilustración 84. Índice de severidad, Fuente autor

En este módulo se encontraron los siguientes errores: 1 error de prioridad media y 1 error de prioridad alta, equivalentes al 40% por ciento de errores en el módulo. Se presentaron errores en la visualización y adjunto de las fotos al momento de realizar la denuncia, así como también problemas al visualizar las denuncias generadas por el ciudadano en el mapa.

Módulo docente

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo Docente, Ciclo 1

Tabla 49. Pruebas de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	tipo de cuenta, correo electrónico, contraseña	Se diligencia los campos de entrada El sistema arrojará un mensaje confirmando si el proceso fue exitoso.	Mensaje que confirma la acción realizada	No
Pruebas Funcionales	Alta	registro de denuncias	Se visualizará una tabla con los campos de entrada, en donde se encuentra todo el listado de las denuncias asignadas	No aplica	No
Pruebas Funcionales	Media	No aplica	Se procederá a elegir la denuncia	Verificación del documento	NO

			asignada por el administrador		
Pruebas Funcionales	Alta	Estados de denuncia	se selecciona los campos de entrada del estado	Generar Estado	SI
Pruebas Funcionales	Baja	Notas	Se visualiza una casilla con la asistencia y tendrá la opción de generar unas notas o comentarios	Mensaje que confirma la acción realizada	SI
Pruebas Funcionales	Alta	Selecciona PDF	Una vez seleccionado la denuncia, descargar el documento PDF diligenciado por el usuario	Reporte de denuncia	NO
Pruebas Funcionales	Baja	datos del docente	Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que	No hay Mensaje	SI

			ha ocurrido un error.		
Pruebas Funcionales	Baja	datos del docente	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje que confirma la acción realizada	no

La siguiente grafica hace referencia a los porcentajes de error de acuerdo con la severidad en el ciclo 1 en el módulo Docente:

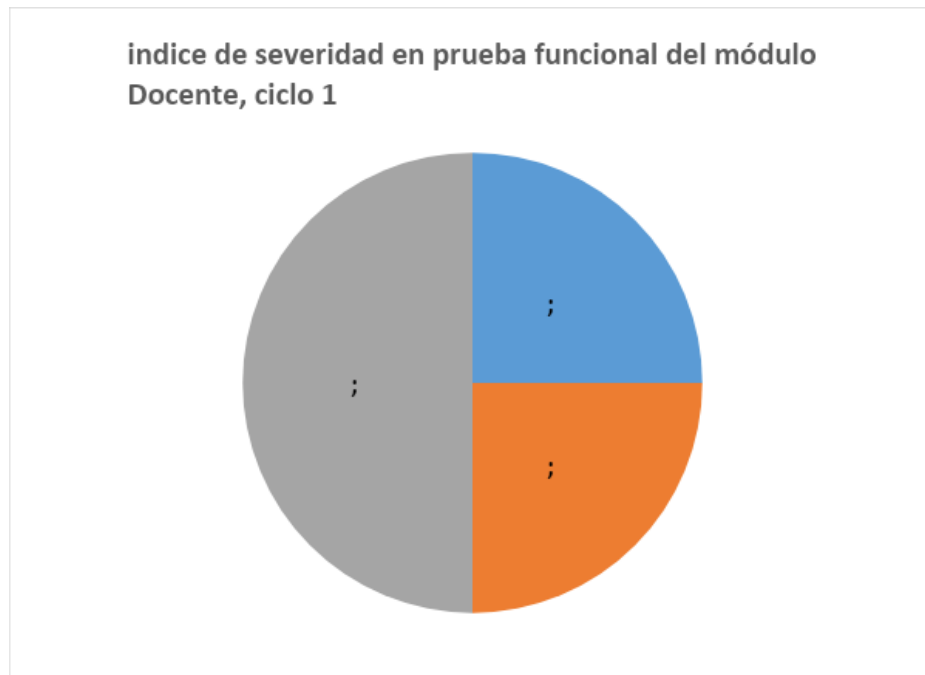


Ilustración 85. Índice de severidad, Fuente autor

En este módulo se encontraron tres errores; 1 error de prioridad alta, 1 error de prioridad baja y 1 error de prioridad media equivalentes al 45% por ciento de errores en el módulo, donde el docente no identifica los estados para cada denuncia, además no se genera mensaje de información donde se evidencien las validaciones de los tipos de datos a llenar en los campos de texto y no se visualizan las notas.

Ciclo 2.0

Las pruebas funcionales realizadas en la versión 1 de estos módulos mostraron ciertas inconsistencias omitidas durante la implementación del diseño del front – end, esto indica que hay errores de prioridades por corregir a nivel visual, se evaluaron 8 ítems considerados para el chequeo y verificación de estos módulos, así mostrando un resultado un poco más claro y detallado en porcentaje.

Teniendo en cuenta los resultados de las pruebas de severidad del ciclo uno de los módulos administrador, ciudadano y docente, se procede a realizar nuevamente las pruebas con una segunda fase para garantizar que los hallazgos en el ciclo uno fue corregidos. A continuación, se detallan las tablas con los resultados de la información de cada uno de los módulos:

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo Administrador, Ciclo 2

Módulo Administrador

Tabla 50. Pruebas de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	Correo electrónico, contraseña, tipo de usuario	Se procede a llenar los campos requeridos para ingresar como usuario admin	Mensaje de confirmación si se realizó la acción.	NO

Pruebas Funcionales	Alta	Nombre, correo electrónico, dirección, cedula, lugar de expedición, código de docente, contraseña	Se procede a llenar los campos requeridos para crear un nuevo docente y hacer el registro en la base de datos	Mensaje de proceso exitoso.	NO
Pruebas Funcionales	Media	Nombre, correo electrónico, dirección, cedula, lugar de expedición, código de docente, contraseña	Muestra en pantalla un modal en donde se actualizan los datos del docente, administrador	Mensaje de proceso exitoso	NO
Pruebas Funcionales	Alta	Asignar docente botón (editar)	Se visualizará un botón este asignara las denuncias a los docente	No hay Mensaje	NO
Pruebas Funcionales	Baja	Datos del Administrador	Cuando se han ingresado datos de entrada y estos	No hay Mensaje	SI

			son incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error		
Pruebas Funcionales	Baja	Datos Entrada	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje de proceso exitoso.	NO

Pruebas Funcionales	Media	Nombre, correo electrónico, dirección, cedula, lugar de expedición, código de docente, contraseña	Muestra en pantalla un botón en donde se actualiza los datos del docente, administrador	Mensaje de proceso exitoso	
					SI

Como veíamos en el ciclo uno de la prueba de severidad realizada en el módulo administrador se halló dos errores donde solo fue corregido el de severidad media

correspondiente a la visualización de mensajes de confirmación al momento de realizar el registro de un docente.

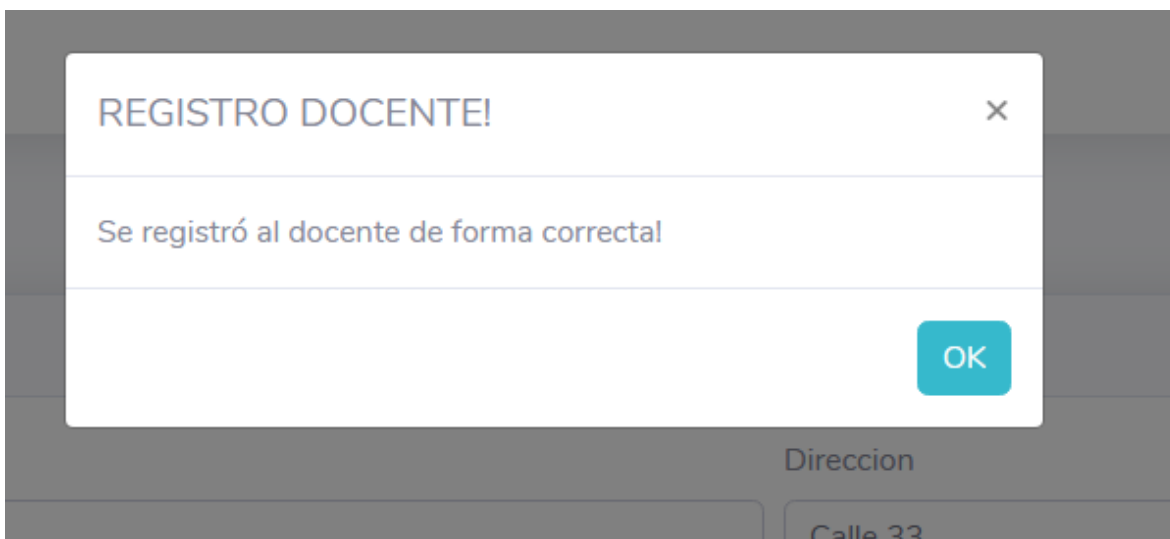
Para la corrección de este error fue necesario crear una vista modal la cual se activará al momento dar clic en el botón “registrar “la estructura de la vista se diseñó de la siguiente manera:

Ilustración 86. Fragmento de código, Fuente autor

```
<div class="modal fade" id="modalRegistroDocente" tabindex="-1" role="dialog" aria-label
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title">REGISTRO DOCENTE!</h5>
        <button type="button" class="close" onclick="actualizar();" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        Se registró al docente de forma correcta!
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-info" onclick="actualizar();">OK</button>
      </div>
    </div>
  </div>
</div>
```

Como se puede observar en el fragmento de código Dando como resultado el modal de la siguiente ilustración se agregó una etiqueta <div> la cual contiene el nombre del modal, el mensaje a mostrar y la confirmación a través del botón “ok “ dando como resultado el modal de la siguiente ilustración:

Ilustración 87. Registro docente, fuente autor



Con respecto al segundo error no se evidencia la corrección en su totalidad por lo que amerita una tercera prueba para determinar su total corrección.

La siguiente grafica hace referencia a los porcentajes de error de acuerdo con la severidad en el ciclo 2 en el módulo Administrador:



Ilustración 88. Índice de severidad, fuente autor

En este módulo se encontró 1 error de prioridad media, que no fue solucionado, esto equivale al 0,3% de la totalidad de errores en el módulo administrador. Se espera que para el siguiente ciclo de pruebas quede reducido el margen de error.

Módulo Ciudadano

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo ciudadano, Ciclo 2

Tabla 51. Pruebas de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error

Pruebas Funcionales	Alta	Correo electrónico, contraseña, tipo de usuario	Se procede a llenar los campos requeridos para ingresar como usuario	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	Alta	Dirección, Agregar ubicación, Agregar imágenes	Se procede a llenar los campos requeridos para crear una nueva denuncia hacer el registro en la base de datos	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	Alta	Check list, control numérico	se diligencia el formulario para generar la denuncia	asignación de datos	NO
Pruebas Funcionales	Alta	Botón mis denuncias, botón mapa de casos, botón de nueva denuncia	Muestra en pantalla todas las funcionalidades donde se realiza la denuncia	información para diligenciar	NO

Pruebas Funcionales	Alta	Botón actualizar mis datos, Botón cambiar contraseñ a	Se visualizará los botones de entrada, en donde se actualice los datos y contraseña del usuario.	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	media	Mapa de casos	Se visualiza en el mapa todas las denuncias realizadas por todos los usuarios.	Denuncias agregadas en el mapa	NO
Pruebas Funcionales	Alta	imágenes	Se adjunta foto de evidencia y soporte al formato	Imágenes de la denuncia	NO
Pruebas Funcionales	Baja	Botón enviar respuesta	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje de confirmación si se realizó la acción.	NO

Como veíamos en el primer ciclo de la prueba de severidad realizada en el módulo ciudadano, se hallaron 2 errores donde fueron corregidos satisfactoriamente, en el cual se trabajó en los hallazgos de severidad alta correspondiente a la visualización del mapa de casos con las denuncias generadas; para solucionar este caso se implantaron las variables “lati” y “long”(latitud y longitud) junto con la variable marker2, la cual posiciona el icono bajo la url: “img/locate.png”, este cuenta con unas dimensiones de 50 x 50 (px) al llamar la función “showPosition” automáticamente crea la referencia en el mapa . En el siguiente fragmento de código se puede apreciar la estructura del código:

```
for(i=0; i<json_data.length; i++){
    point = json_data[i];
    var lati = getLat(point.coordenadas);
    var long = getLon(point.coordenadas);
    lati = parseFloat(lati);
    long = parseFloat(long);
    var pos = {lat:lati, lng:long};
    var marker2 = new google.maps.Marker({position: pos, map: mapas,
        title: 'Uluru (Ayers Rock)',
        label:{
            text:i,
            color:'white'
        },
        icon:{
            url:'img/Locate.png',
            scaledSize: new google.maps.Size(50, 50),
            origin: new google.maps.Point(0,0), // origin
            anchor: new google.maps.Point(0, 0)
        }
    });
};
```

Como resultado se obtuvo la referenciarían en el mapa tal y como lo ilustra la siguiente imagen:



Ilustración 89. Mapa de casos, Fuente autor

Para la corrección de los errores en la visualización de las imágenes se creó un array el cual valida si la casilla de adjuntar imagen se encuentra vacía, si es de tipo imagen (JPG, PNG, TIF), estas imágenes son almacenadas en la carpeta “Denuncia” localizada en el servidor.

```
while ($imgD=mysqli_fetch_array($query)){
    $json = $imgD['imagenes'];
    if($json==null){
        echo "No contiene adjuntos";
    }else{
        $obj = json_decode($json);
        echo "<div class='container'>
            <div class='row'>
                <div class='col'>
                    N#1
                    <img src='../denuncias/imagenes/" . $obj->{'img1'} . "' width='100%' >
                </div>
                <div class='col'>
                    N#2
                    <img src='../denuncias/imagenes/" . $obj->{'img2'} . "' width='100%'>
                </div>
            </div>
            <br />
            <div class='row'>
                <div class='col'>
                    N#3
                    <img src='../denuncias/imagenes/" . $obj->{'img3'} . "' width='100%'>
                </div>
            </div>
        </div>";
    }
```

Una vez realizada esta acción, los datos son mostrados a través de la opción “Adjuntos” como se puede apreciar en la siguiente vista:

Descargar	Adjuntos	Fecha
Generar Reporte		2019-11-05 18:46:22
Generar Reporte		2019-11-05 16:01:27
Generar Reporte		2019-11-05 15:39:02

Ilustración 90. Vista de adjuntos, fuente autor

Al dar clic en la opción de “Adjuntos”, como resultado se obtienen los datos almacenados dando solución al hallazgo del ciclo 1, mostrándose de la siguiente manera:

Imágenes

x

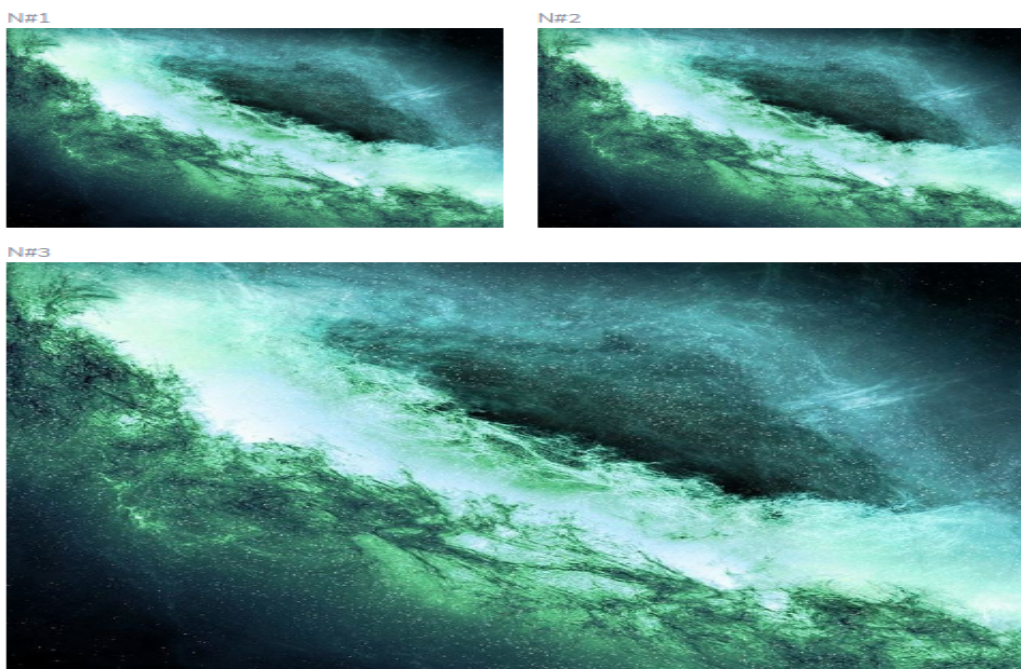


Ilustración 91. Evidencias fotográficas, fuente autor

Módulo Docente

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo Docente, Ciclo 2

Tabla 52, Pruebas de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
----------------	-----------	---------------	---------------	----------------------	-------

Pruebas Funcionales	Alta	tipo de cuenta, correo electrónico, contraseña	Se diligencia los campos de entrada El sistema arrojará un mensaje confirmando si el proceso fue exitoso.	Mensaje que confirma la acción realizada	NO
Pruebas Funcionales	Alta	registro de denuncias	Se visualizará una tabla con los campos de entrada, en donde se encuentra todo el listado de las denuncias asignadas	No aplica	NO
Pruebas Funcionales	Media	No aplica	Se procederá a elegir la denuncia asignada por el administrador	Verificación del documento	NO
Pruebas Funcionales	Alta	Estados de denuncia	se selecciona los campos de entrada de los estados	Generar Estado	NO

Pruebas Funcionales	Baja	Notas	Se visualiza una casilla con la asistencia y tendrá la opción de generar unas notas o comentarios fotos	Mensaje que confirma la acción realizada	NO
Pruebas Funcionales	Alta	Selecciona PDF	Una vez seleccionado la denuncia, descargar el documento PDF diligenciado por el usuario	Reporte de denuncia	NO
Pruebas Funcionales	Baja	datos del docente	Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No hay Mensaje	SI
Pruebas Funcionales	Baja	datos del docente	Se muestra un mensaje cuando se ha	Mensaje que confirma la acción realizada	NO

			cumplido una operación.		
--	--	--	-------------------------	--	--

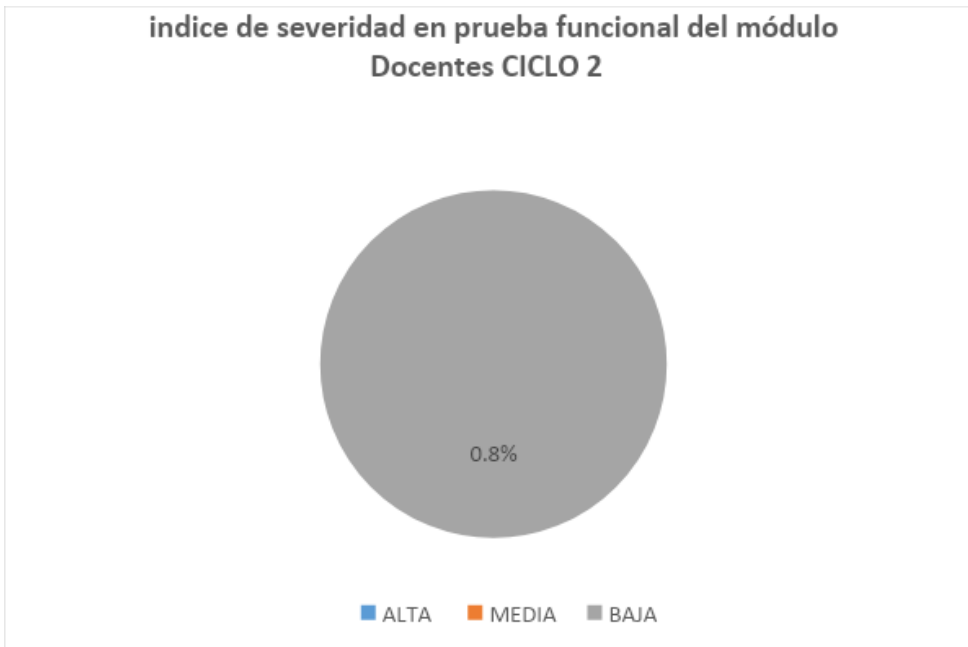
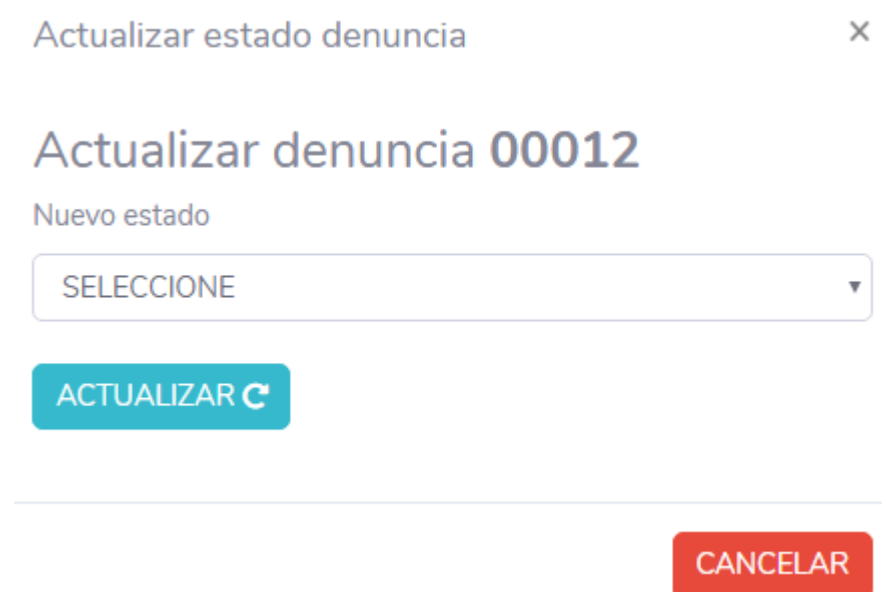


Ilustración 92. Índice de severidad de prueba, fuente autor

Como veíamos en el ciclo uno de la prueba de severidad realizada en el módulo docente, se hallaron tres errores donde fueron corregidos de la siguiente manera: Para la solución del error de severidad alta correspondiente al manejo de estados de las denuncias se agregó un div el cual contiene las etiquetas <option> con cada uno de los estados solicitados por el cliente, quedando de la siguiente manera:

```
<div class="modal-body">
  <h3>Actualizar denuncia <b><span id="id-denuncia"></span></b></h3>
  <label for="select-nuevo-estado">Nuevo estado</label>
  <SELECT class="form-control" id="select-nuevo-estado">
    <option>SELECCIONE</option>
    <option>EN ESTUDIO</option>
    <option>EN TRAMITE</option>
    <option>PRESENTADA</option>
    <option>FINALIZADA</option>
    <option>RECHAZADA</option>
```

Implementada esta estructura se obtuvo la siguiente vista:



Actualizar estado denuncia ×

Actualizar denuncia 00012

Nuevo estado

SELECCIONE ▼

ACTUALIZAR ↻

CANCELAR

Ilustración 93. Actualizar estado de la denuncia, fuente autor

El docente puede cambiar el estado de la denuncia dependiendo la gestión que este realice y de acuerdo con las políticas de consultorio jurídico.

Para actualizar el estado de la denuncia se creó la siguiente estructura dentro del archivo `actualizar_estado_denunica.php`

```

<?php
if(isset($_POST['nuevo_estado']) && isset($_POST['id'])) {

    $idDenuncia = $_POST['id'];
    $nuevo_estado = $_POST['nuevo_estado'];

    require("../unilab_clases/complaint_class.php");

    $Complaint = new Complaint ();

    $result = $Complaint->updateState($idDenuncia,$nuevo_estado);

    echo($result);
}
?>

```

Al gestionar las denuncias realizadas por el ciudadano los docentes cuentan la característica “estado de denuncia” la cual se visualiza de la siguiente manera:



Gestion de denuncias	
ID denuncia	Estado denuncia
00012	EN ESTUDIO  

Ilustración 94. Gestión de denuncia, fuente autor

También se corrigió una severidad baja, corresponde a agregar notas





Estado denuncia	Notas	Descargar reporte
EN ESTUDIO ● 	 	 Generar Reporte

Ilustración 95. Estado de la denuncia, fuente autor

Las notas son agregadas por medio del método POST

```

<?php
require("../unilab_clases/complaint_class.php");
$denuncia = new Complaint();

if(isset($_POST['id']) && isset($_POST['nota'])){
    $nota = $_POST['nota'];
    $id = $_POST['id'];

    if($denuncia->noteIsEmpty($id)){
        echo($denuncia -> addNote($id,$nota,true)); //addNote(nota,first)
    }else{
        echo($denuncia -> addnote($id,$nota,false));
    }
    // echo($id.", ".$nota);
}

```

Como resultado se obtuvo la siguiente ilustración donde el usuario docente puede agregar las notas relacionadas a la gestión del caso; las notas van siendo generadas en filas lo que permite tener una trazabilidad da cada una de ellas

A screenshot of a modal window titled "Agregar nota" with a close button (X) in the top right corner. Below the title is the label "Detalles de la nota:" followed by a large, empty text input field. At the bottom of the modal, there are two buttons: "Cancelar" (grey) and "Guardar" (blue).

Ilustración 96. Agregar nota, fuente autor

Al consultar las notas se muestra una ventana modal con la información de cada una de estas tal y como lo muestra la siguiente ilustración:

A screenshot of a modal window titled "Notas de la denuncia" with a close button (X) in the top right corner. The main content of the modal is the text "Notas para la denuncia 00012". At the bottom right of the modal, there is a blue button labeled "OK".

Ilustración 97. Notas de la denuncia, fuente autor

Por último, se resolvió severidad media la cual describe que no se generan mensajes de información donde se evidencien las validaciones de los tipos de datos a llenar en los campos de texto. Par la solución de este error se implementó una estructura

utilizando la siguiente sentencia de código en la cual se utilizó una función llamada "datos" donde se validan los campos de texto, mostrando así la letra de campos vacíos o campos incompletos:

```
    },  
    success: function(datos){  
        if(datos=="vacío"){  
            $("#etiResultadoEditDocente").html("<div class='alert alert-danger' role='alert'>ERROR! no  
        }else{  
            if(datos==false){  
                $("#etiResultadoEditDocente").html("<div class='alert alert-danger' role='alert'>ERROR! !
```

Como resultado se obtuvo la siguiente vista donde se aplica la validación si el campo se encuentra vacío, o si las contraseñas coinciden o no:

Cambiar contraseña ciudadano ×

Contraseña Antigua

Nueva contraseña

Repetir nueva contraseña

No coinciden las contraseñas

ACTUALIZAR ↻

docente@gmail.com

Ilustración 98. Cambiar contraseña, fuente autor

Ciclo 3.0

Teniendo en cuenta los resultados de las pruebas de severidad del ciclo uno y dos de los módulos administrador, ciudadano y docente, se procede a realizar nuevamente las pruebas con una tercera y última fase para garantizar que los hallazgos en el ciclo uno y dos se hallan corregidos. A continuación, se detallan las tablas con los resultados de la información de cada uno de los módulos:

Tabla de resultados de pruebas de severidad por prioridad baja media y alta del módulo Administrador, Ciclo 3

Módulo Administrador

Tabla 53. Ciclo 3, prueba de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	Correo electrónico, contraseña, tipo de usuario	Se procede a llenar los campos requeridos para ingresar como usuario admin	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	Alta	Nombre, correo electrónico, dirección, cedula, lugar de expedición, código de docente, contraseña	Se procede a llenar los campos requeridos para crear un nuevo docente y hacer el registro en la base de datos	Mensaje de proceso exitoso.	NO
Pruebas Funcionales	Media	Nombre, correo electrónico, dirección, cedula, lugar de expedición,	Muestra en pantalla un modal en donde se actualizan los datos del	Mensaje de proceso exitoso	NO

		código de docente, contraseña	docente, administrador		
Pruebas Funcionales	Alta	Asignar docente botón (editar)	Se visualizará un botón este asignará las denuncias a los docente	No hay Mensaje	NO
Pruebas Funcionales	Baja	Datos del Administrador	Cuando se han ingresado datos de entrada y estos son incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error	No hay Mensaje	NO
Pruebas Funcionales	Baja	Datos Entrada	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje de proceso exitoso.	NO
Pruebas Funcionales	Media	Nombre, correo electrónico,	Muestra en pantalla un botón en	Mensaje de proceso exitoso	

		dirección, cedula, lugar de expedición, código de docente, contraseña	donde se actualiza los datos del docente, administrador		
--	--	---	---	--	--

Módulo Ciudadano

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	Correo electrónico, contraseña, tipo de usuario	Se procede a llenar los campos requeridos para ingresar como usuario	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	Alta	Dirección, Agregar ubicación, Agregar imágenes	Se procede a llenar los campos requeridos para crear una nueva denuncia hacer el registro	Mensaje de confirmación si se realizó la acción.	NO

			en la base de datos		
Pruebas Funcionales	Alta	Check list, control numérico	se diligencia el formulario para generar la denuncia	asignación de datos	NO
Pruebas Funcionales	Alta	Botón mis denuncias, botón mapa de casos, botón de nueva denuncia	Muestra en pantalla todas las funcionalidades donde se realiza la denuncia	información para diligenciar	NO
Pruebas Funcionales	Alta	Botón actualizar mis datos, Botón cambiar contraseña	Se visualizará los botones de entrada, en donde se actualice los datos y contraseña del usuario.	Mensaje de confirmación si se realizó la acción.	NO
Pruebas Funcionales	media	Mapa de casos	Se visualiza en el mapa todas las denuncias realizadas por todos los usuarios.	Denuncias agregadas en el mapa	NO
Pruebas Funcionales	Alta	imágenes	Se adjunta foto de evidencia y	Imágenes de la denuncia	NO

			soporte al formato		
Pruebas Funcionales	Baja	Botón enviar respuesta	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje de confirmación si se realizó la acción.	NO

Módulo Docente

Tabla 54. Ciclo 3, prueba de severidad, fuente autor

Tipo de Prueba	Severidad	Datos Entrada	Procedimiento	Salidas (Resultados)	Error
Pruebas Funcionales	Alta	tipo de cuenta, correo electrónico, contraseña	Se diligencia los campos de entrada El sistema arrojará un mensaje confirmando si el proceso fue exitoso.	Mensaje que confirma la acción realizada	NO
Pruebas Funcionales	Alta	registro de denuncias	Se visualizará una tabla con	No aplica	NO

			los campos de entrada, en donde se encuentra todo el listado de las denuncias asignadas		
Pruebas Funcionales	Media	No aplica	Se procederá a elegir la denuncia asignada por el administrador	Verificación del documento	NO
Pruebas Funcionales	Alta	Estados de denuncia	se selecciona los campos de entrada de los estados	Generar Estado	NO
Pruebas Funcionales	Baja	Notas	Se visualiza una casilla con la asistencia y tendrá la opción de generar unas notas o comentarios fotos	Mensaje que confirma la acción realizada	NO
Pruebas Funcionales	Alta	Selecciona PDF	Una vez seleccionado la denuncia, descargar el documento	Reporte de denuncia	NO

			PDF diligenciado por el usuario		
Pruebas Funcionales	Baja	datos del docente	Cuando se han ingresado datos incorrectos se muestra un mensaje de advertencia explicando que ha ocurrido un error.	No hay Mensaje	NO
Pruebas Funcionales	Baja	datos del docente	Se muestra un mensaje cuando se ha cumplido una operación.	Mensaje que confirma la acción realizada	NO

Durante la verificación de la aplicación “Gestor Urbano”, se logra inicialmente eliminar el margen de errores que existían en los diferentes módulos que conforman aplicación (ciudadano, administrador y docente), seguido a esto se logra comprobar que dicha aplicación cumple con cada uno de los requerimientos funcionales establecidos.

Cabe resaltar que al momento de realizar la validación de la aplicación; se realizaron una serie de pruebas de funcionalidad, lo que facilito la identificación del cumplimiento de los requerimientos funcionales pactados con el cliente (consultorio Jurídico Unisinu), el cual se basa en la gestión de denuncias de control urbano en obras de construcción.

Finalmente podemos concluir con los resultados obtenidos de las gráficas evidenciadas de todos los módulos se puede decir que la aplicación tuvo un 100% por ciento de funcionalidad, obtenidos en los ciclos 1,2, por consiguiente, se cumplió con el objetivo esperado según los requerimientos funcionales de la aplicación.

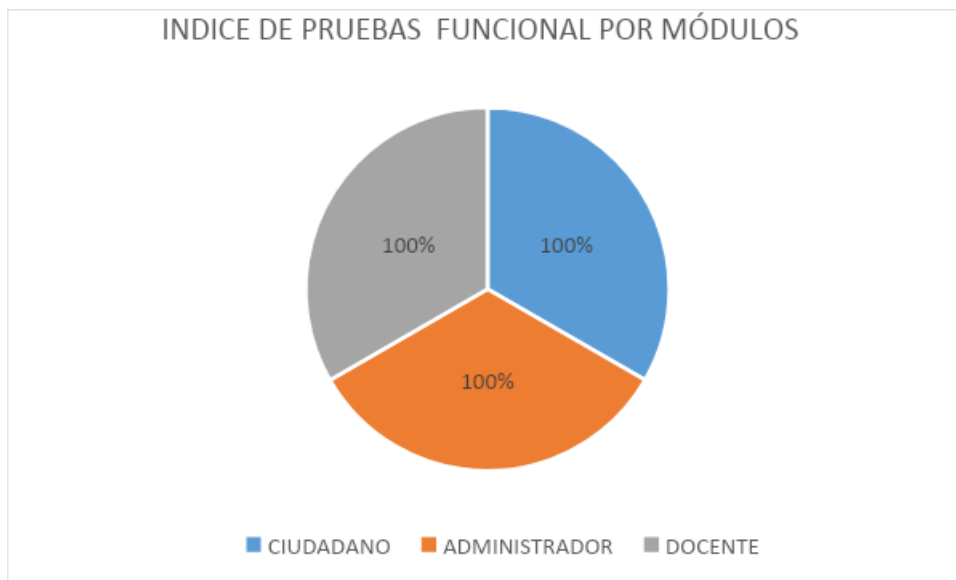


Ilustración 99. Índice de pruebas funcionales por módulos, fuente autor

6. CONCLUSIONES Y RECOMENDACIONES

Este proyecto permitió la construcción de un prototipo para de una aplicación web móvil, que le facilite al ciudadano realizar procesos de denuncia sobre el control urbano de obras en construcción en la localidad tres de Cartagena. Para lograr dicho objetivo

fue necesario iniciar con una fase de análisis que corresponde al primer objetivo específico del proyecto. En esta fase se describe el cronograma del proyecto, se definieron los agentes involucrados que son relevantes en la aplicación, se describió el plan de conocimientos y el plan habilidades, de igual forma se estableció el alcance del proyecto, realizando a detalle el levantamiento de la información con el cliente (Consultorio jurídico de la Universidad del Sinú).

Como resultado de esta fase se definieron tres (3) módulos: Ciudadano, Administrador y Docente. Así mismo se señalaron las características generales y específicas de la aplicación según los módulos establecidos. Cabe resaltar que también se tuvo en cuenta la definición de las suposiciones y dependencias dentro del alcance, señalando entonces los requisitos funcionales y no funcionales del sistema a detalle con cada una de las características de determinado módulo. Para esto se tuvo en cuenta los requisitos del cliente, cada característica fue sometida a su aprobación para proceder con la siguiente etapa del proyecto. Este proceso se realizó mediante una metodología de entrevistas con el cliente, en las cuales se levantaron las respectivas minutas para dejar por sentado cada compromiso y requerimiento.

Siguiendo la metodología CMMI se procedió con la fase de diseño en la cual se realizaron los diseños UML. Estos diseños son los diagramas de secuencia, de proceso, de componentes, y de despliegue, casos de uso, diagramas entidad/ relación y el modelo relacional, de esta forma se representó cada requisito del sistema. Una vez establecidos y diseñados estos diagramas, se diseñaron los Mockups, usando el formato REQ_VIEW_V1, el cual permite evidenciar cada uno de los componentes de los Mockups.

Seguidamente se realizó la codificación conforme al diseño planteado, se desarrolló un prototipo de aplicación web responsive usando en lenguaje de programación PHP. El desarrollo de este prototipo se implementó bajo el modelo de desarrollo rápido de aplicaciones (RAD), se utilizó la versión 7.2.7 de PHP y JavaScript para la codificación del Backend, se utilizaron las métricas de desarrollo de acuerdo al manual PHP, de tal forma que este fuera fácil de leer e interpretar. Además de esto se definieron los estilos

de la aplicación utilizando CSS y Bootstrap para el desarrollo Fronted, como resultado final se obtuvo un software intuitivo, ágil y adaptable a cualquier dispositivo móvil o PC.

Posteriormente se determinó un escenario de pruebas web, las cuales permitieron validar los requerimientos y requisitos del sistema planteados por el usuario, aplicándolos en la localidad tres de Cartagena. Esto se llevó a cabo bajo el dominio “gestorurbano.com” en el cual , junto con el cliente (Consultorio jurídico de la Universidad del Sinú) se realizaron las pruebas funcionales y de interfaz de acuerdo a los requisitos del sistema, diseño planteado y el alcance del proyecto, todo esto con el fin de evidenciar las posibles fallas y errores en el rendimiento del sistema y poder mitigar el número de inconsistencias detectadas, de tal manera, que a la hora de que el software sea utilizado por el usuario final, este cuente con un buen diseño, sea intuitivo y adaptable a las necesidades del usuario, sea eficiente y de buena calidad. Cada una de estas pruebas o validaciones se realizaron por ciclos y en cada ciclo se generó un informe acerca de los resultados obtenidos, llevando a cabo cada una de las pautas establecidas por este, dando como resultado la entrega exitosa del producto a satisfacción del cliente (Consultorio jurídico de la Universidad del Sinú).

Para trabajo futuro recomendamos tener en cuenta los siguientes aspectos:

- Crear módulo para la creación de perfiles en el sistema.
- Integrar la aplicación en conjunto con el plan de ordenamiento territorial (POT) para la identificación de zonas de riesgo a través de la geolocalización.
- Permanecer al día con el pago de los servicios de hosting y dominio de la aplicación, al igual que los servicios de Google maps Platform.
- Implementar el servicio de notificaciones por correo electrónico para alertar sobre el vencimiento de las denuncias

- Mejorar la función del chat para las condiciones de horario de atención, de tal manera que los ciudadanos tengan el conocimiento de los horarios de respuesta del consultorio jurídico.
- Implementar nueva característica en el sistema de tal manera que se puedan agregar documentos a la denuncia, en la gestión del docente
- Implementar nueva característica que permita que las fotos que son tomadas o cargadas por el ciudadano puedan visualizarse en el mapa de casos.
- Implementar nueva característica, que permita colocar dinámicamente Hipervínculos en el mapa para redireccionar a YouTube, de tal manera que el ciudadano pueda ver el informe de la denuncia de manera audiovisual.
- Permitir que las denuncias se les puedan cargar fotos y videos a la vez de que estos archivos puedan ser actualizados por el administrador y docente; de esta manera se pretende mostrar un antes y después en cada caso reportado.
- Implementar nueva funcionalidad que permita al administrador mirar la trazabilidad de los estados de las denuncias, de tal manera que, si el docente cambia de un estado a otro, se pueda evidenciar si lo hizo durante el tiempo estipulado para la respuesta, para efecto de indicadores.
- Implementar funcionalidad para que el administrador pueda modificar el Timer.
- Implementar funcionalidad que permita al administrador activar o desactivar las denuncias vencidas.

- Implementar funcionalidad que permita al administrador, listar a los usuarios ciudadanos registrados; de tal manera que, en caso de falta a las normas y políticas de la aplicación, el administrador pueda inactivar a dicho usuario.

7. REFERENCIAS

[1] Diario el Heraldó . Afectados por caída del edificio Blas de Lezo

[2] R. D. Rodríguez, 06 de mayo de 2017..

[3] J. Montañó, 6 de mayo 2017 , pp. , p. 003, . .

- [4] A. d. j. R. Licona, « «Diseño y desarrollo de una aplicación Móvil para dispositivos android para un sistema de alerta temprano en los arroyos de barranquilla,,» » Barranquilla..
- [5] J. C. A. Avilez, « «Diseño e implementación de una aplicación móvil android para la sistematización del estudio, la prevención y control de enfermedades en los cultivos de algodón como herramienta de aprendizaje,,» » Programa de ingeniería agronómica , córdoba, córdoba.
- [6] J. R. C. Alvarado, ««Desarrollo de una aplicación para dispositivos móviles, que permita administrar pedidos y controlar rutas de los vendedores,,» » JUAN JURIS CÍA LTDA. división de perfumería.
- [7] N. Carr, «The Glass Cage,» *Revista Semana*, 2014.
- [8] «Patrón MVC en java con netbeans,,» » Mouse Blogspot. .
- [9] ». Alex Rodriguez, ««Aprender programación Java desde cero,,» » pp. Programar.com, vol. 1, nº 84..
- [10] Guia esencial:Las bases de datos dan soporte a las tendencias de TI,Searchdata
- [11] search data center, ««Guia Esencial de base de datos»»..».
- [12] «. Neoattack, «concepto de Framework».
- [13] www.abc.es, «Concepto de API».
- [14] Google.com, « Concepto de Google Maps».
- [15] ,. C. E. P. Prado, « «Backend Developer i Python, SQL,,» » DevCode.com. .
- [16] .. L. Chart, «El Lenguaje unificado de modelado UML».
- [17] A. E. López, «Consultas en bases de datos».
- [18] About español. , «Significado y definición de el término aplicación Móvil».
- [19] Plataforma estatal de docentes de tecnología, , ««Tecnología»».
- [20] Metodologia Rad.weebly.com, , ««Modelo de Desarrollo Rad»»..».
- [21] P. Virtuales.
- [22] MegaPractical.com, , «CMMI significado y uso».
- [23] pdcahome.com, , ««CMMI»»..».

[24] S. Díaz, 2009. .

ANEXOS

Anexo A: Gestion de Requetimienos (RD)

Matriz de trazabilidad

Nombre del Proyecto:		DESARROLLO DE UN PROTOTIPO DE APLICACIÓN WEB MÓVIL PARA DENUNCIAS DE CONTROL URBANO DE OBRAS EN CONSTRUCCIÓN EN LA LOCALIDAD TRES DE LA CIUDAD DE CARTAGENA, APLICANDO BUENAS PRÁCTICAS DE INGENIERÍA DE SOFTWARE (CMMI)		
Líder del Proyecto:	Deimer Avila Argote			
Fecha:				

Control de Versiones				
Nombre del Documento	Versión	Fecha	Autor/Aprobación	Comentarios
Matriz de Trazabilidad	1.0		Luis Blanquicet	Actualización de requisitos del sistema

ID	Módulo	Descripción	Prioridad (Alta, Media, Baja)	Nivel de Completado	Observaciones
RF001	Módulo Ciudadano	Permitir al usuario registrarse en la aplicación como nuevo usuario del sistema.	Alta	100%	
RF002	Módulo Ciudadano	. Permitir a los usuarios ingresar en el sistema utilizando login su contraseña, y	Media	100%	

		recibiendo luego de una autenticación positiva de la APP.			
RF003	Módulo Ciudadano	Permite a cada usuario modificar sus datos personales modificables, como su contraseña, teléfono, correo electrónico.	Alta	100%	
RF004	Módulo Ciudadano	Permitir al usuario (ciudadano) diligenciar el cuestionario para la identificación de la denuncia, así agregarla al sistema.	Media	100%	
RF005	Módulo Ciudadano	Permite al usuario decidir el manejo de la denuncia	baja	100%	
RF006	Módulo Ciudadano	Permitir al usuario (ciudadano) agregar imágenes del sitio de la incidencia.	Alta	100%	
RF007	Módulo Ciudadano	Permitir al usuario (ciudadano) agregar el punto	Alta	100%	

		geolocalizado del sitio de la incidencia.			
RF008	Módulo Administrador	Permitir al usuario Administrador ver las incidencias reportadas por el ciudadano y a quien se le fue asignada, por el equipo de trabajo (docentes, estudiantes)	Alta	100%	
RF009	Módulo Administrador	Permitir al usuario Administrador las incidencias reportadas por el ciudadano asignarlas al equipo de trabajo (docentes, estudiantes)	Alta	100%	
RF010	Módulo Administrador	Permitir al usuario Administrador ver el histórico de denuncias reportadas por los ciudadanos	Alta	100%	
RF011	Módulo Administrador	Permitir al sistema de la aplicación ingresar el usuario registrándose	Media	100%	

		como nuevo usuario del sistema.			
RF012	Módulo Administrador	Permitir al usuario (administrador) hacer modificaciones a las preguntas el cuestionario para la denuncia.	Alta	100%	
RF013	Módulo Docente	Permitir al usuario registrarse en la aplicación como nuevo usuario del sistema.	Alta	100%	
RF014	Módulo Docente	Permitir al usuario (Docente) ver las incidencias asignadas por el Administrador.	Media	100%	
RF015	Módulo Docente	Permitir al usuario (administrador) hacer un cuadro estadístico donde se refleje las denuncias recientes.	Media	100%	
RF016	Módulo Docente	Permitir al usuario (administrador) visualizar y contestar todas las	Media	100%	

		preguntas o inquietudes de los ciudadanos.			
--	--	--	--	--	--

Solicitud de cambio de requerimientos

Nombre del Proyecto:	DESARROLLO DE UN PROTOTIPO DE APLICACIÓN WEB MÓVIL PARA DENUNCIAS DE CONTROL URBANO DE OBRAS EN CONSTRUCCIÓN EN LA LOCALIDAD TRES DE LA CIUDAD DE CARTAGENA, APLICANDO BUENAS PRÁCTICAS DE INGENIERÍA DE SOFTWARE (CMMI)
Líder del Proyecto:	Deimer avila

Administración del Cambio							Estimación de Tiempo		Compromiso Afectado			Seguimiento a los Cambios		
Nº	ID Cambio	Fecha Solicitud	Fase de Desarrollo que Afecta	Documento Afectado	Descripción del Cambio	Tipo de Cambio	Fecha Inicio	Fecha Fin	Documento Afectado	Compromiso Afectado	Aprobado Por	El cambio fue Realizado a la Fecha	Fecha Real	Revisado Por
1	1R	07/16/2019	Fase de Análisis de Requerimientos	Especificación de Requerimientos (SRS)	Se modificaron los casos de usos para el módulo	Requerimiento nuevo	08/21/2019	08/30/2019	Especificación de Requerimientos (SRS)	Se agregó la funcionalidad de visualizar las por	Alo nso cor tina	Si	08/30/2019	Deimer Avila - Líder de Pro

					administrador, docente para funcionalidad de timer					colores cuando una denuncia está por vencer				yec to
2	2R	07/16/2019	Fase de Análisis de Requerimientos	Validación de Requerimientos Funcionales	Se agregaron requisitos técnicos del sistema	Mejora	08/28/2019	08/28/2019	Validación de Requerimientos Funcionales	Se agregó la funcionalidad de poder visualizar todas las denuncias generadas sin importar el tipo de usuario	Alo nso cor tina	Si	08/28/2019	Dei me r Avila - Líder de Proyecto
3	3R	07/16/2019	Fase de Conceptualización	Alcance Detallado del Producto	Se agregaron las nuevas funcionalidades que tendrán los módulos:	Requerimiento nuevo	07/26/2019	22/02/2019	Especificación de Requerimientos (SRS)	Se agregó la funcionalidad en el estado de la denuncia colocando	Alo nso cor tina	Si	08/28/2019	Dei me r Avila - Líder de Proyecto

					admi nistra dor, docen te y ciuda dano					nuev os esta dos				
--	--	--	--	--	--	--	--	--	--	---------------------------	--	--	--	--