



**INSTITUTO TECNOLÓGICO DE CD. GUZMÁN**

**TITULACIÓN INTEGRAL  
TESIS**

**TEMA:  
IMPLEMENTACIÓN DE TECNOLOGÍAS EN MANTENIMIENTO A  
INFRAESTRUCTURA**

**QUE PARA OBTENER EL TÍTULO DE:  
INGENIERO INFORMÁTICO**

**PRESENTA:  
DAVID VELASCO LUJÁN**

**ASESORA:  
DRA. RAQUEL OCHOA ORNELAS**

**CD.GUZMÁN, JALISCO, MÉXICO. SEPTIEMBRE DE 2020**



“2020, Año de Leona Vicario, Benemérita Madre de la Patria”

Cd. Guzmán, Municipio de Zapotlán el Grande, Jal. **12/10/2020**

**ASUNTO:** Liberación de Proyecto para Titulación integral

**M.C. FAVIO REY LÚA MADRIGAL**  
**JEFE DE LA DIVISION DE ESTUDIOS PROFESIONALES**  
PRESENTE

Por este medio informo que ha sido liberado el siguiente proyecto para la Titulación integral:

|                             |                            |
|-----------------------------|----------------------------|
| <b>Nombre del Egresado</b>  | <b>DAVID VELASCO LUJÁN</b> |
| <b>Carrera:</b>             | Ingeniería Informática     |
| <b>No. De Control</b>       | <b>15290768</b>            |
| <b>Nombre del proyecto:</b> | “PATCH-UP-REPORT”          |
| <b>Producto</b>             | TESIS                      |

Agradezco de antemano su valioso apoyo en esta importante actividad para la formación profesional de nuestros egresados.

ATENTAMENTE  
*Innova, Transforma y Crea para ser Grande*

**M.S.C. OMAR CRISTIAN VARGAS GONZÁLEZ**  
**JEFE DEL DEPTO. SISTEMAS Y COMPUTACIÓN**



|                               |  |                                |
|-------------------------------|--|--------------------------------|
| Dra. Raquel Ochoa Ornelas<br> | Dra. María Guadalupe Sánchez Cervantes<br> | Dr. Daniel Fajardo Delgado<br> |
| Nombre y Firma del Asesor     | Nombre y Firma del Revisor                 | Nombre y Firma del Revisor     |

MIAH/CRM/OCVG/mgoc\*



## Agradecimientos

*A mi madre, Verónica: por ser un apoyo incondicional y estar siempre acompañándome; en las desveladas y desmañadas. Siempre pendiente de mí.*

*A mi padre, David: por su esfuerzo, trabajo, comprensión y apoyo durante todos estos años.*

*A mis hermanas, Sarahí y Belén: por estar siempre presentes y dentro de las posibilidades de cada una, brindarme su apoyo absoluto.*

*A mis abuelas y abuelos: Alejandra, Reyes, María y Ricardo; por siempre creer en mí, ser un digno ejemplo de vida y llevarme en sus corazones.*

*A mi asesora y maestra, la Dra. Raquel Ochoa: por todos los conocimientos brindados dentro y fuera del salón de clases.*

*¡Que nadie se quede fuera! Gracias a toda mi gran familia y amigos, que, aunque no estén mencionados explícitamente, siempre estuvieron ahí para ayudarme de alguna manera u otra.*

## Resumen

El control del mantenimiento correctivo a la infraestructura ya sea pública o privada es complejo, pues involucra temas administrativos, financieros y de seguimiento a las condiciones de las vialidades, ecología, sistema de drenaje, agua potable y áreas verdes entre otros. El desarrollo de este proyecto promueve el uso de las Tecnologías de la Información y Computación (TIC) facilitando la administración del proceso y seguimiento al registro de reportes realizados por los usuarios, así como el control administrativo para atender las peticiones de manera oportuna administrando la asignación de herramientas y recursos humanos que se requieran. Además, se mantiene un registro permanente en un control de seguimiento de reportes atendidos a través de una bitácora para su posterior análisis.

El proyecto comprende el desarrollo de una plataforma Web y Móvil enfocada a apoyar las labores de mantenimiento a la infraestructura pública en el Departamento de Obras Públicas en el Ayuntamiento Municipal de Zapotlán el Grande. La aplicación móvil está dirigida a los ciudadanos para que por medio de ésta puedan realizar cualquier tipo de reporte y concentrarlo en una base de datos en la nube que a su vez será consultada por la aplicación Web en las oficinas del propio Ayuntamiento. El usuario puede realizar reportes, así como consultarlos de manera general o por ubicación. Estas dos aplicaciones trabajan en conjunto ofreciendo información en tiempo real a todos los usuarios.

Para el desarrollo del proyecto se utilizaron diferentes tecnologías como *MPAndroidChart*, *Servicios de Google*, *Recycler View*, *Glide*, *Material Design*, *Retrofit*, *Firebase*, *Volley*, *Jitpack.io*, *Bootstrap* y *Libphone number* entre otras. Se utilizó *MySQL* para el manejo de la Base de Datos en un servidor en la nube, así como diferentes lenguajes y frameworks como *Android*, *HTML*, *PHP*, *JavaScript* y *Java*.

# Índice

|   |     |
|---|-----|
| Agradecimientos.....                        | i   |
| Resumen .....                               | ii  |
| Índice de figuras .....                     | vii |
| Índice de tablas .....                      | xi  |
| Capítulo 1. Introducción.....               | 1   |
| Capítulo 2. Antecedentes.....               | 2   |
| Capítulo 3. Objetivos.....                  | 3   |
| 3.1 Objetivo general .....                  | 3   |
| 3.2 Objetivos específicos.....              | 3   |
| Capítulo 4. Hipótesis .....                 | 4   |
| Capítulo 5. Justificación .....             | 5   |
| Capítulo 6. Marco teórico.....              | 6   |
| 6.1 HTML 5.....                             | 6   |
| 6.1.1 Funcionamiento de HTML5.....          | 6   |
| 6.1.2 Estructura.....                       | 7   |
| 6.1.3 Sintaxis y corrección de errores..... | 8   |
| 6.2 CSS3 .....                              | 8   |
| 6.2.1 Funcionamiento .....                  | 8   |
| 6.2.2 CSS y HTML.....                       | 9   |
| 6.2.3 Selectores.....                       | 10  |
| 6.2.4 Selectores de clase.....              | 10  |
| 6.2.5 Selectores de identificador.....      | 11  |
| 6.3 JavaScript .....                        | 12  |
| 6.4 PHP.....                                | 12  |
| 6.4.1 Funcionamiento de PHP .....           | 13  |
| 6.4.2 Estructura de un script PHP.....      | 13  |
| 6.4.3 Variables.....                        | 15  |
| 6.5 MySQL .....                             | 16  |
| 6.5.1 Queries en MySQL.....                 | 16  |
| 6.5.2 Acceso a base de datos desde PHP..... | 17  |

|   |    |
|---|----|
| 6.6 JSON.....                                 | 18 |
| 6.6.1 Sintaxis JSON.....                      | 18 |
| 6.7 BootStrap.....                            | 19 |
| 6.7.1 Características.....                    | 20 |
| 6.7.2 Implementación en HTML.....             | 20 |
| 6.7.3 Componentes.....                        | 21 |
| 6.8 Java.....                                 | 26 |
| 6.8.1 Funcionamiento.....                     | 27 |
| 6.8.2 Java para Android.....                  | 27 |
| 6.9 Android Studio.....                       | 29 |
| 6.9.1 Arquitectura.....                       | 29 |
| 6.9.2 Vistas (Views).....                     | 29 |
| 6.9.3 Layout.....                             | 31 |
| 6.9.4 Preferencias (Shared Preferences).....  | 31 |
| 6.9.5 Intención (Intent).....                 | 32 |
| 6.9.6 Permisos.....                           | 33 |
| 6.9.7 MySQL desde Android Studio.....         | 36 |
| 6.10 MP AndroidChart.....                     | 37 |
| 6.11 Servicios de Google (Play Services)..... | 37 |
| 6.11.1 Mapas.....                             | 37 |
| 6.11.2 Localización.....                      | 38 |
| 6.12 RecyclerView.....                        | 39 |
| 6.12.1 Uso.....                               | 39 |
| 6.13 Glide.....                               | 41 |
| 6.13.1 Uso.....                               | 42 |
| 6.14 Material Design.....                     | 43 |
| 6.14.1 Animación de actividades.....          | 43 |
| 6.15 Firebase.....                            | 44 |
| 6.15.1 Google Analytics.....                  | 45 |
| 6.16 Volley.....                              | 45 |
| 6.16.1 Consumo servicios Web.....             | 46 |
| 6.17 JitPack.io.....                          | 48 |

|  |     |
|--|-----|
| 6.18 Libphone number.....                                | 49  |
| Capítulo 7. Metodología.....                             | 51  |
| 7.1 Análisis de requerimientos .....                     | 51  |
| 7.1.1 Entidades Externas .....                           | 51  |
| 7.1.2 Procesos.....                                      | 52  |
| 7.1.3 Almacenes de datos .....                           | 52  |
| 7.2 Diseño.....  | 52  |
| 7.2.1 Diagrama de caso de uso general .....              | 52  |
| 7.2.2 Diagrama de Warnier/Orr para el Administrador..... | 54  |
| 7.2.3 Diagrama de Warnier/Orr para el usuario .....      | 55  |
| 7.2.4 Diagrama de Warnier/Orr para el empleado.....      | 55  |
| 7.2.5 Diagrama de Clases .....                           | 56  |
| 7.2.6 Diagramas de Compras.....                          | 57  |
| 7.2.7 Diagramas de Reportes.....                         | 60  |
| 7.2.8 Diagramas de Bajas de herramientas.....            | 63  |
| 7.2.9 Diagramas de Devoluciones de Herramientas.....     | 66  |
| 7.3 Pruebas .....  | 69  |
| 7.3.1 Caso de prueba: Movimiento Compras .....           | 69  |
| 7.3.2 Caso de prueba: Baja de Herramientas.....          | 71  |
| 7.4 Pruebas en la aplicación móvil.....                  | 73  |
| 7.5 Implantación.....                                    | 75  |
| Capítulo 8. Resultados.....                              | 76  |
| 8.1 Sistema Web.....                                     | 76  |
| 8.1.1 Administrador.....                                 | 77  |
| 8.1.2 Empleado.....                                      | 86  |
| 8.2 Aplicación móvil .....                               | 88  |
| 8.2.1 Usuario .....                                      | 88  |
| Capítulo 9. Temas de discusión.....                      | 95  |
| 9.1 Comunicación entre plataformas Web y móvil .....     | 95  |
| 9.2 Diseño responsivo móvil .....                        | 97  |
| 9.3 Imágenes desde servidor sin ayuda de librerías.....  | 100 |
| Capítulo 10. Conclusiones.....                           | 102 |

|  |     |
|--|-----|
| Capítulo 11. Referencias bibliográficas..... | 103 |
| Anexos.....                                  | 105 |
| Adapter.java.....                            | 105 |
| activity_resultados.xml.....                 | 107 |
| Proveedores.php .....                        | 110 |

## Índice de figuras

|   |    |
|---|----|
| Figura 1. Peticiones a servidor de archivo en formato HTML.....                                     | 6  |
| Figura 2. Estructura básica de un documento HTML. ....  | 7  |
| Figura 3. Peticiones a servidor de archivo en formato HTML con hoja de estilos CSS incorporada..... | 9  |
| Figura 4. Evolución y unificación de HTML y CSS.....  | 10 |
| Figura 5. Selectores de clase en CSS y su uso en HTML.....  | 11 |
| Figura 6. Selectores de clase en CSS.....   | 11 |
| Figura 7. Funcionamiento de PHP con HTML.....   | 13 |
| Figura 8. Script PHP en documento HTML.....   | 14 |
| Figura 9. Resultado del script php en documento html.....   | 15 |
| Figura 10. Variables en PHP.....  | 15 |
| Figura 11. Resultado del Query.....   | 17 |
| Figura 12. Query de actualización de la tabla herramientas.....                                     | 17 |
| Figura 13. Conexión a base de datos.....  | 18 |
| Figura 14. Objeto JSON.....   | 19 |
| Figura 15. Arreglo JSON.....  | 19 |
| Figura 16. Implementación de bootstrap en HTML.....   | 20 |
| Figura 17. Grid con ejemplo de estructuración.....  | 21 |
| Figura 18. Código de una barra de navegación usando bootstrap.....                                  | 22 |
| Figura 19. Ejecución de barra de navegación.....  | 22 |
| Figura 20. Código de un input con bootstrap.....  | 23 |
| Figura 21. Ejecución de caja de texto.....  | 23 |
| Figura 22. Código de un dropdown (select) con bootstrap.....  | 24 |
| Figura 23. Ejecución de dropdown.....   | 24 |
| Figura 24. Código de un botón con bootstrap.....  | 25 |
| Figura 25. Ejecución de botón.....  | 25 |
| Figura 26. Código principal de una ventana modal.....   | 25 |
| Figura 27. Ejecución de ventana modal.....  | 26 |
| Figura 28. Funcionamiento de Java.....  | 27 |
| Figura 29. Código fuente de una clase en Java.....  | 28 |
| Figura 30. Archivo .class.....  | 28 |
| Figura 31. Código XML de una vista.....   | 30 |
| Figura 32. Ejecución del código XML en Android.....   | 30 |
| Figura 33. Código XML de Layout.....  | 31 |
| Figura 34. Guardando correo y contraseña con Preferencias.....                                      | 32 |
| Figura 35. Intento interno para abrir activity.....   | 32 |
| Figura 36. Intento externo para enviar correo.....  | 33 |
| Figura 37. Permisos en el archivo android manifest.....   | 33 |
| Figura 38. Solicitar permisos peligrosos.....   | 35 |
| Figura 39. Solicitud al usuario de permiso peligroso.....   | 35 |

|   |    |
|---|----|
| Figura 40. Acceso a base de datos desde Android Studio.....                     | 36 |
| Figura 41. Uso de gráfica de barras con la biblioteca Mp AndroidChart.....      | 37 |
| Figura 42. Creación de credencial para uso de mapas de google.....              | 38 |
| Figura 43. Archivo con el API de mapas.....                                     | 38 |
| Figura 44. Uso de servicio de localización.....                                 | 39 |
| Figura 45. RecyclerView.....  | 40 |
| Figura 46. Vista de RecyclerView.....   | 40 |
| Figura 47. Visualización de una consulta con RecyclerView.....                  | 41 |
| Figura 48. Usando CircleView.....   | 42 |
| Figura 49. Uso de Glide.....  | 42 |
| Figura 50. Habilitando transiciones.....  | 43 |
| Figura 51. Uso de transiciones.....   | 44 |
| Figura 52. Uso de Google Analytics (firebase).....                              | 45 |
| Figura 53. Creación de servicio Web con PHP.....                                | 46 |
| Figura 54. Consulta de servicio Web desde el navegador.....                     | 47 |
| Figura 55. Consumo de servicio Web en Android Studio.....                       | 47 |
| Figura 56. Uso de LibPhone en Java.....   | 49 |
| Figura 57. LibPhone desde la Web.....   | 50 |
| Figura 58. Diagrama de caso de uso general.....                                 | 53 |
| Figura 59. Diagrama de Warnier/Orr para el Administrador.....                   | 54 |
| Figura 60. Diagrama de Warnier/Orr para el Usuario.....                         | 55 |
| Figura 61. Diagrama de Warnier/Orr para el Empleado.....                        | 55 |
| Figura 62. Diagrama de clases.....  | 56 |
| Figura 63. Diagrama de caso de uso expandido para registrar Compra.....         | 57 |
| Figura 64. Diagrama de flujo de datos para Registrar Compra.....                | 57 |
| Figura 65. Diagrama de colaboración para el caso de uso Registrar Compra.....   | 58 |
| Figura 66. Diagrama de estados para Compras.....                                | 59 |
| Figura 67. Diagrama de caso de uso expandido para Reportes.....                 | 60 |
| Figura 68. Diagrama de flujo de datos del movimiento Reportes.....              | 60 |
| Figura 69. Diagrama de colaboración para Registrar Reportes.....                | 61 |
| Figura 70. Diagrama de estados para Registrar Reportes.....                     | 62 |
| Figura 71. Diagrama de caso de uso expandido para bajas de herramientas.....    | 63 |
| Figura 72. Diagrama de flujo de datos para registrar bajas de herramientas..... | 63 |
| Figura 73. Diagrama de colaboración para bajas de herramientas.....             | 64 |
| Figura 74. Diagrama de estados para Registrar Bajas.....                        | 65 |
| Figura 75. Diagrama de caso de uso expandido para Registrar Devoluciones.....   | 66 |
| Figura 76. Diagrama de flujo de datos para Devoluciones de Herramientas.....    | 66 |
| Figura 77. Diagrama de colaboración para Devoluciones de Herramientas.....      | 67 |
| Figura 78. Diagrama de estados para la Devolución de Herramientas.....          | 68 |
| Figura 79. Interfaz de la prueba del movimiento Compras.....                    | 70 |
| Figura 80. Interfaz de la prueba baja de herramientas.....                      | 72 |
| Figura 81. Interfaz de la prueba Registrar Reporte.....                         | 74 |
| Figura 82. Página principal del sistema Web.....                                | 76 |

|   |    |
|---|----|
| Figura 83. Menú Administrador - Catálogos.....                | 77 |
| Figura 84. Menú Administrador - Movimientos. ....             | 77 |
| Figura 85. Menú Administrador - Consultas.....                | 78 |
| Figura 86. Catálogo de Usuarios. ....                         | 78 |
| Figura 87. Editar estatus de Usuarios.....                    | 79 |
| Figura 88. Catálogo de Proveedores.....                       | 79 |
| Figura 89. Registrar Proveedor.....                           | 80 |
| Figura 90. Movimiento Compras.....                            | 80 |
| Figura 91. Movimiento Devolución de Herramientas.....         | 81 |
| Figura 92. Movimiento de Reportes. ....                       | 81 |
| Figura 93. Movimiento Bajas de Herramienta .....              | 82 |
| Figura 94. Consulta de Compras por rango de fechas.....       | 82 |
| Figura 95. Consulta de Compras por proveedor.....             | 83 |
| Figura 96. Consulta de Compras por personal que autoriza..... | 83 |
| Figura 97. Consulta de Devoluciones por trabajo. ....         | 84 |
| Figura 98. Consulta de Bajas por fechas. ....                 | 84 |
| Figura 99. Consulta de Bajas por trabajos.....                | 85 |
| Figura 100. Consulta de Bajas por personal que autoriza ..... | 85 |
| Figura 101. Vista principal del Empleado.....                 | 86 |
| Figura 102. Menú - movimientos Empleado.....                  | 86 |
| Figura 103. Movimiento Reportes empleado. ....                | 87 |
| Figura 104. Pantalla de inicio. ....                          | 88 |
| Figura 105. Pantalla de registro.....                         | 89 |
| Figura 106. Pantalla de registro 2.....                       | 89 |
| Figura 107. Permiso cámara .....                              | 89 |
| Figura 108. Permiso ubicación.....                            | 89 |
| Figura 109. Permiso archivos. ....                            | 89 |
| Figura 110. Menú principal 2.....                             | 90 |
| Figura 111. Menú principal 1. ....                            | 90 |
| Figura 112. Resumen del reporte.....                          | 90 |
| Figura 113. Realizar reporte .....                            | 90 |
| Figura 114. Consulta general.....                             | 91 |
| Figura 115. Consulta en mapa .....                            | 91 |
| Figura 116. Resultados del reporte .....                      | 92 |
| Figura 117. Actualizar información.....                       | 92 |
| Figura 118. Pantalla contacto 2. ....                         | 93 |
| Figura 119. Pantalla contacto. ....                           | 93 |
| Figura 120. Equipo. ....                                      | 93 |
| Figura 122. Cambio a inglés.....                              | 94 |
| Figura 121. Idioma .....                                      | 94 |
| Figura 123. Método HTTP Handler .....                         | 95 |
| Figura 124. Archivo API.....                                  | 96 |
| Figura 125. Uso de Volley.....                                | 97 |

|  |     |
|--|-----|
| Figura 126. Uso de Adobe XD.....                                 | 97  |
| Figura 127. Dos layouts.....                                     | 98  |
| Figura 128. Diseño para pantallas con anchura mayor a 500dp..... | 98  |
| Figura 129. Menú para pantallas pequeñas. ....                   | 99  |
| Figura 130. Imagen en base64.....                                | 100 |
| Figura 131. Imagen sin base64.....                               | 100 |
| Figura 132. Mostrando imagen con librería Glide. ....            | 101 |

## Índice de tablas

|  |    |
|--|----|
| Tabla 1. Prueba del movimiento Compras .....               | 70 |
| Tabla 2. Prueba del movimiento baja de herramientas.....   | 72 |
| Tabla 3. Prueba del movimiento Reporte .....               | 73 |
| Tabla 4. Cronograma de implantación de la plataforma ..... | 75 |

## Capítulo 1. Introducción

El control de la infraestructura ya sea en instalaciones físicas o en equipos que requieran mantenimiento en el sector público o privado complejo, ya que involucra asuntos administrativos, recursos financieros y de seguimiento a las condiciones de la infraestructura en cualquier ámbito ya sea público o privado.

La plataforma fue desarrollada para dos escenarios, una aplicación Web de uso administrativo en las oficinas de los ayuntamientos y una aplicación móvil donde los usuarios realizan sus reportes y consultan el seguimiento correspondiente.

Este proyecto permite obtener la colaboración permanente de los ciudadanos para reportar y atender de manera sistemática problemas de mantenimiento a infraestructura. Además, elimina riesgos al contar con una inspección permanente de las instalaciones que puedan poner en peligro la seguridad, el medio ambiente o patrimonio, mejorando además la calidad de los procesos o servicios.

El desarrollo de este proyecto facilita la administración del proceso y seguimiento al registro de reportes realizados por los usuarios, así como el control administrativo para atender las peticiones de manera oportuna, administrando la asignación de herramientas y recursos humanos que se requieran.

A lo largo de la presente tesis, se podrán ver los objetivos generales y específicos, hipótesis, trabajos previos, metodología, así como las diferentes tecnologías implementadas para el desarrollo de esta plataforma Web y móvil.

## Capítulo 2. Antecedentes

Estudios de la Organización de las Naciones Unidas (ONU) indican que el 54,6% de la población del planeta, viven en ciudades y pronostica que para el 2050 el 70% vivirá en ciudades, colocando a las personas en el centro de desarrollo, utilizando las TIC. De esta manera, se construyen ciudades competitivas, atractivas al turismo y resilientes, con procesos planificados que promueven la participación ciudadana y desarrollo sostenible e integrado (Bouskela et. al, 2016).

Existen algunas aplicaciones en el mercado informático que ofrecen reportes vía celular por parte de los ciudadanos, sin embargo, las aplicaciones son básicas y no consideran el seguimiento y notificaciones. Algunas de estas aplicaciones se describen a continuación:

- **Bache - 24:** Este programa tiene por objeto garantizar la seguridad vial y peatonal de los habitantes de Ciudad de México mediante un servicio de bacheo de alta presión o bacheo en frío. A través de un sistema de captura, monitoreo y supervisión se da seguimiento a las solicitudes de bacheo, el cual concluye al informar mediante un reporte peticionario en un tiempo menor a 24 horas.
- **072 Móvil:** Es una aplicación al servicio de la Secretaría de Obras y Servicios de la Ciudad de México. A través de ella se pueden reportar baches y fugas de agua. Esta se encuentra disponible para descargarse de forma gratuita en los sistemas operativos Android y iOS.
- **Fracttal:** Gestión de activos y mantenimiento para un mundo conectado. Tiene distintos planes de pago, el básico por 195 dólares, empresarial 458 dólares y corporativo 994 dólares. Esta aplicación funciona en la industria de servicios, hotelera, salud, minería, alimentos, transporte, tecnología, ciudades, construcción y recreación.

El proyecto desarrollado considera una plataforma Web de administración y seguimiento instalada dentro del propio H. Ayuntamiento de Zapotlán el Grande.

## Capítulo 3. Objetivos

A continuación, se plantea el objetivo general y los objetivos específicos de esta tesis.

### 3.1 Objetivo general

Facilitar el control en el proceso administrativo y seguimiento al mantenimiento en la infraestructura de gobierno, instituciones o empresas mediante una plataforma Web y móvil que permita monitorear y atender de manera eficiente los reportes emitidos por los usuarios.

### 3.2 Objetivos específicos

- Gestionar reportes de usuarios a través de una aplicación móvil.
- Administrar compras de herramientas y asignación de personal.
- Registrar bajas y devoluciones de herramientas.
- Disponer de una bitácora en la Web de seguimiento a mantenimiento de infraestructura.
- Consultar el estado de los reportes en tiempo real.
- Generar diversos informes como el de inventario, compras y reportes en formato PDF para la impresión y/o análisis de este.
- En el ámbito público, la ciudadanía podrá hacer reportes de infraestructura del municipio, incrementando la supervisión y participación ciudadana.

## Capítulo 4. Hipótesis.

Para poder cumplir con los objetivos de la presente tesis, se planteó la siguiente hipótesis:

“Al utilizar un sistema multiplataforma en dependencias públicas o privadas que permita monitorear en tiempo real los reportes emitidos por usuarios en la atención del mantenimiento y seguimiento a infraestructura, reducirá costos y mejorará la calidad en los procesos y servicios, así como la toma de decisiones”.

## Capítulo 5. Justificación

En el sector público y en el sector privado es importante aplicar de manera constante una política de mantenimiento adecuada a la infraestructura, reduciendo inconvenientes, mejorando el manejo racional de capital humano y financiero, así como la prolongación de la vida útil de los equipos o infraestructuras físicas.

Se realizaron visitas al H. Ayuntamiento Municipal de Zapotlán el Grande, específicamente en el departamento de Obras Públicas para proponer un proyecto que permitiera supervisar de manera eficiente todo el proceso de mantenimiento a la infraestructura de la ciudad, la cual incluye mantenimiento de calles, alumbrado, parques y jardines, así como el mantenimiento en los servicios de agua potable. El responsable estuvo de acuerdo en el desarrollo del proyecto, integrando los requerimientos y necesidades que como institución pública les interesaba. Además, solicitaron llevar un control en el personal que realiza cada mantenimiento y un inventario de herramientas que se utilizan en las tareas asignadas.

El proceso anteriormente lo realizaban de manera manual, además no disponían de una supervisión eficiente en las calles para monitorear todos los problemas por atender. De acuerdo con los requerimientos se consideró también el desarrollo de una aplicación móvil que permitiera a los ciudadanos reportar cualquier desperfecto en la ciudad, con la finalidad de que fuera atendido de manera inmediata.

Este proyecto no se limita a ser utilizado en el sector público, también se ha hablado con empresarios que están interesados en implementar la aplicación en hoteles, industria cementeras y aguacateras.

Finalmente, al llevar un control eficiente en todo el proceso de mantenimiento a la infraestructura, la información se encuentra concentrada en una base de datos, la cual permitirá consultas y estadísticas para su posterior análisis y mejor toma de decisiones en las inversiones que se realicen a futuro en el tema de infraestructura.

## Capítulo 6. Marco teórico

En este capítulo, se presentan las principales tecnologías que se integran en el presente proyecto.

### 6.1 HTML 5

HTML5 es la quinta y última versión, cuyas siglas significan lenguaje de marcado de hipertexto (por sus siglas en inglés). Es considerado el producto de la combinación de tres tecnologías: HTML, CSS y JavaScript. Estas tecnologías son altamente dependientes una de la otra, aun cuando son autónomas (Gauchat, 2012). Este lenguaje es un estándar que funciona como referencia del software que conecta con la elaboración de páginas Web en sus distintas versiones, con esto se define la estructura básica con el código (en HTML) para la definición de contenido de una página Web, para en ella poder poner elementos como párrafos de texto, títulos, subtítulos, imágenes, viñetas, tablas, videos, contenido multimedia, entre otros.

HTML es un lenguaje de etiquetas orientado a la Web o un “Idioma que entiende la computadora y procesa para dar respuesta”. Lo único necesario para escribirlo es un editor de texto- Este se escribe en forma de etiquetas, rodeadas por corchetes angulares (< y >), también haciendo uso de la diagonal para cerrar una etiqueta (Muñoz, 2012).

#### 6.1.1 Funcionamiento de HTML5

En la Figura 1 se puede ver la sucesión de pasos cuando se solicita una página en formato HTML a través de cualquier navegador. Este proceso inicia cuando un cliente realiza la petición al servidor, lo que se hace a través de una dirección del tipo `http://.../nombrearchivo.html`. Después de esto, el servidor lee del disco duro esa página, la devuelve al navegador y la página en formato HTML se muestra al cliente.

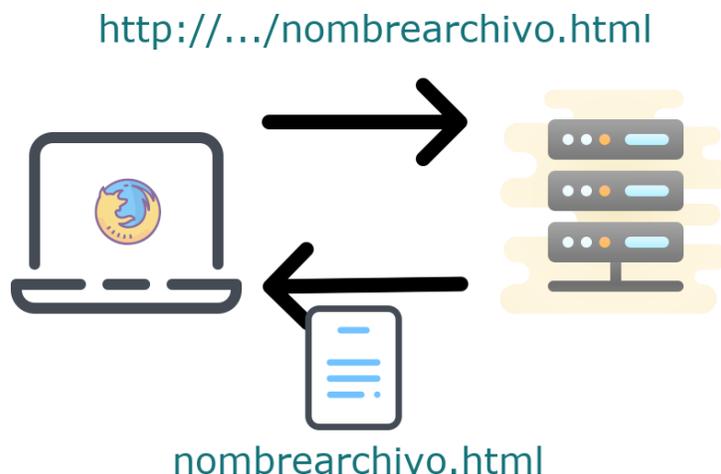


Figura 1. Peticiones a servidor de archivo en formato HTML

## 6.1.2 Estructura

Los documentos en formato HTML se deben encontrar estrictamente organizados. Cada parte del documento está debidamente marcada por las etiquetas específicas correspondientes. Estas etiquetas envolverán al resto del código que esté dentro de ellas. (Gauchat, 2012).

En la Figura 2 se muestran algunos ejemplos básicos de etiquetas en HTML. Como se puede ver, estos son conceptos simples, existen otras etiquetas con diferentes funcionalidades como `<link>` para usar archivos de estilo en CSS, `<img>` para insertar una imagen, `<figure>` y `<figcaption>` para ayudar a ser mucho más específicos a la hora de declarar contenido en el documento.

```

Estructura básica de un documento HTML

<html lang="es"> Comienzo del documento en español

<head> Cabecera del documento
    <title></title> Texto que aparecerá en el título de la ventana
</head>

<body> Cuerpo del documento
    <table> Tabla
        <tr>
            <th>Marca</th> Título de la columna
        </tr>
        <tr>
            <td>HP</td> Contenido de la celda
        </tr>
    </table>

    <h1></h1> Etiqueta para títulos.

    <p></p> Párrafo

    <footer></footer> Pie de página

</body>

</html> Indica el final del documento
```

Figura 2. Estructura básica de un documento HTML.

### 6.1.3 Sintaxis y corrección de errores

Cuando se creó HTML era un tanto difícil la creación de documentos en este, ya que es un lenguaje poco estricto y como consecuencia, los documentos creados resultaban un tanto caóticos, ya que no siempre se cumplía con la sintaxis. Esto hizo que los navegadores Web se hicieran más complejos para poder leer errores y ambigüedades.

HTML5 no define únicamente cómo se deben de analizar los documentos, sino también cómo deben ser interpretados por la máquina cuando la sintaxis de esta no es válida o es ambigua. En la actualidad, los navegadores corrigen algunos de los errores de sintaxis de distinta manera. Es por eso que HTML5 trata de poner fin a esa necesidad de ingeniería inversa de los navegadores, que cada uno a su manera, compite por ver cómo se corrigen errores. (Keith, 2010).

## 6.2 CSS3

CSS son hojas de estilo en cascada que trabajan junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, estilo, fuente, fondo, bordes, entre otros tantos (Mariño, 2015).

Desde hace 25 años ésta tecnología ha tenido una evolución en el tiempo, que actualmente se encuentra en su versión 3. Pero su función sigue siendo la misma que al principio: dar diseño a las páginas Web. Su nombre en inglés es **Cascading Style Sheets** que en español es «Hojas de estilo en cascada». Las cuales tienen este significado:

- **Cascading:** Los elementos que contiene una página Web se propagan a los elementos que esta abarca.
- **Style:** Lo que se hace con CSS es aplicar estilos visuales a los diferentes elementos que contiene la página Web.
- **Sheets:** Porque el archivo es una hoja con extensión `.css`, la cual se añade al documento HTML de manera general.

### 6.2.1 Funcionamiento

En la Figura 3 se puede observar el funcionamiento de CSS con HTML. El proceso inicia cuando el cliente hace la petición al servidor con una URL, que contiene la dirección del archivo HTML que, a su vez, tiene incorporado CSS. Ejemplo: a través del navegador Mozilla Firefox un usuario hace la petición ingresando la dirección `http://.../nombrearchivo.html`. El servidor buscará el archivo en el disco duro, una vez que lo haya encontrado detectará que tiene incorporado una hoja de estilos con extensión `.css`. Le aplicará los estilos determinados por la hoja de estilos y lo devolverá al usuario para que este

lo pueda ver ya con los diseños aplicados. Es decir, para el usuario se tratará de un solo documento, no de dos o más.

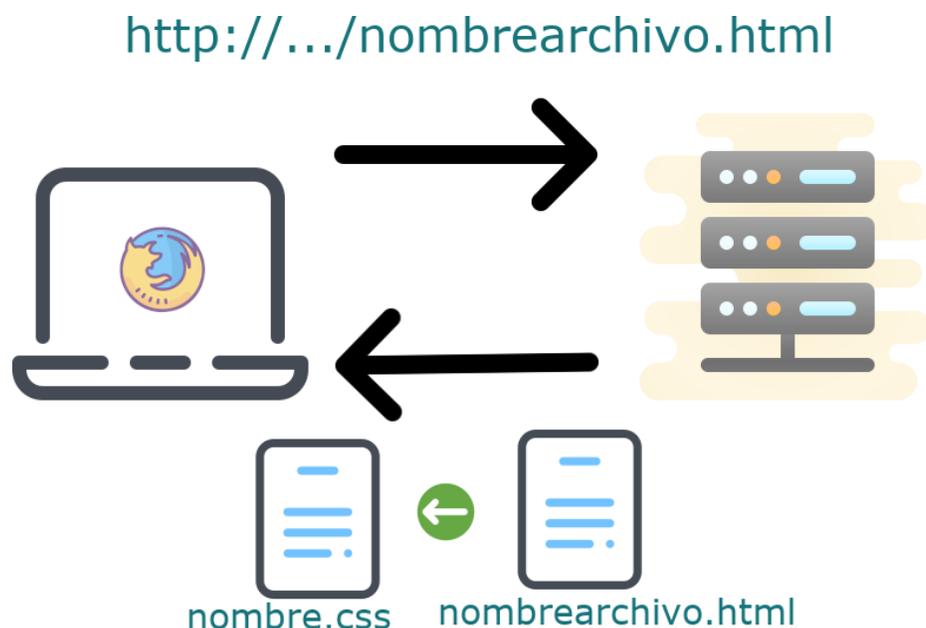


Figura 3. Peticiones a servidor de archivo en formato HTML con hoja de estilos CSS incorporada.

## 6.2.2 CSS y HTML

Para dar funcionalidad y estructura a una página Web y también diseño se necesita de CSS. Dentro de HTML tiene una gran relevancia, aunque oficialmente CSS no tiene nada que ver con HTML5.

CSS fue creado como un complemento para superar las limitaciones y complejidad de HTML, ya que el lenguaje de marcado de hipertexto, por sí mismo no pudo satisfacer las necesidades de los diseñadores. La integración entre estos dos (HTML y CSS) es ahora vital para el desarrollo Web (Gauchat, 2012).

Estos dos han tenido una gran evolución. Partiendo del principio de simplificación y unificación de los dos lenguajes. En la Figura 4 se puede ver un pequeño ejemplo de la evolución que han tenido CSS y HTML.

## Evolución de HTML y CSS

**De :**

```
<link href="nombrHoja.css" rel="stylesheet" type="text/css">
```

**A:**

```
<link href="nombrHoja.css" rel="stylesheet">
```

*Figura 4. Evolución y unificación de HTML y CSS.*

### 6.2.3 Selectores

Para identificar una etiqueta HTML se usan selectores. El selector indica a qué etiqueta de HTML se debe aplicar ese formato.

Para hacer esto simplemente se usa la etiqueta sin los corchetes (<>). Es decir, únicamente el texto de la etiqueta y enseguida de esto, la declaración se efectúa entre dos llaves. Ejemplo:

```
h1 { text-align: center; color:red; }
```

Lo que hará es que todas las etiquetas h1 (que corresponden a un título) estarán alineadas al centro y tendrán un color rojo.

### 6.2.4 Selectores de clase

Hay ocasiones en que no se requiere aplicar el mismo estilo a todas las etiquetas, sino solo a algunas de un tipo en específico. Es por ello por lo que las hojas de estilo cuentan con las clases (class) que permiten al diseñador definir sus propios estilos para determinadas etiquetas (Mouriño, 2015).

Para ello se necesita definir el nombre de la clase precedida de un punto, de ahí se abren llaves y se definen los estilos que se necesitan.

En la Figura 5 se puede ver un ejemplo del uso de clases en css y su implementación en HTML. Lo que se hace es que donde se use la clase ejemplo1 y haya un texto, este se alineará al centro y estará de color rojo.

## Selectores de clase en CSS y su implementación en HTML

**En CSS:**

```
.clase_ejemplo{ text-align: center; color:red; }
```

**Implementación de clase en HTML:**

```
<h1 class="clase_ejemplo">Título con diseño</h1>
```

```
<p class="clase_ejemplo">Texto con diseño</p>
```

*Figura 5. Selectores de clase en CSS y su uso en HTML.*

### 6.2.5 Selectores de identificador

Su uso es muy parecido al de un selector de clase, con la diferencia de que se recomienda usar cuando en el documento se implemente el estilo definido con menor frecuencia que una clase.

El nombre del identificador se precede por un signo de gato (#) y seguido de la declaración de estilo.

En la Figura 6 se puede ver un ejemplo del uso de selectores de identificador y su uso en HTML. Este hará que todas las etiquetas que tengan el id definido en CSS cambien con el estilo definido; en este caso es un texto alineado al centro y de color rojo.

## Selectores en identificador y su implementación en HTML

**En CSS:**

```
#identificador_ejemplo{ text-align: center; color:red; }
```

**Implementación de identificador en HTML:**

```
<p id="identificador_ejemplo">Texto con diseño</p>
```

*Figura 6. Selectores de clase en CSS.*

## 6.3 JavaScript

JavaScript nació hace 25 años con el nombre «LiveScript» y es un lenguaje interpretado basado en guiones que son incorporados directamente en el código HTML. El código es de la parte del cliente para que él lo interprete al hacer la petición para cargar la página. Una de las principales desventajas es que con este lenguaje no pueden crearse programas independientes (Gómez, 2005).

Las principales características de este lenguaje de programación son las siguientes:

- Es multiplataforma.
- Es lenguaje de alto nivel.
- Es un lenguaje interpretado.
- Admite programación estructurada.
- Basado en objetos.
- No se requiere compilación.
- No necesita entorno de desarrollo.
- Maneja la mayoría de los eventos que pueden ocurrir en una página Web.

Una de las principales diferencias entre Java y JavaScript es que con la segunda no se pueden crear interfaces propias para los programas, se necesitan frameworks como React o formularios HTML a través de manejadores de eventos.

## 6.4 PHP

PHP es un lenguaje interpretado del lado del servidor que se caracteriza por su versatilidad, robustez y modularidad. Los sistemas escritos en PHP son integrados directamente al código HTML y ejecutados por un servidor Web a través de un intérprete. Es un lenguaje open source (de código abierto), por lo que es totalmente accesible de forma gratuita para cualquier usuario de la Web.

Es un lenguaje multiplataforma, por lo que los programas tienen que funcionar de la misma manera aún si éstas son distintas. Además de estar preparado para trabajar con más de veinte tipos diferentes de bases de datos. (Gómez, 2005).

### 6.4.1 Funcionamiento de PHP

Se puede ejecutar el código PHP que inserte el resultado en una página Web con contenido HTML siempre y cuando su extensión sea `.php`. Además, se puede controlar qué código se muestre (o no) mediante HTML. Un ejemplo simple es que, si un usuario está logueado en su cuenta, se muestra resultado diferente de uno que no lo está. En la Figura 7 se muestra el funcionamiento de PHP con HTML.

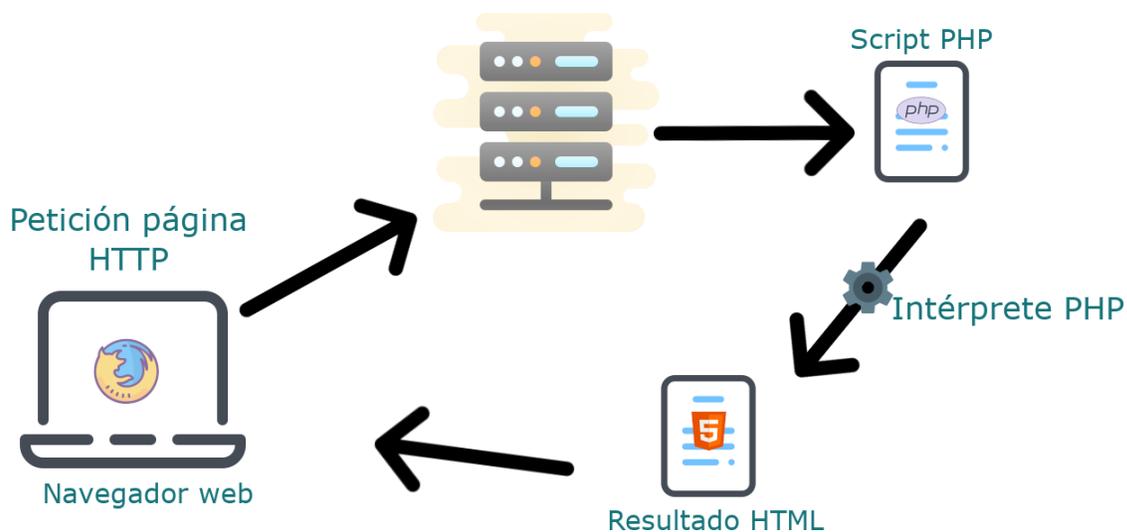


Figura 7. Funcionamiento de PHP con HTML.

### 6.4.2 Estructura de un script PHP

Una página en PHP se crea como si fuera una página HTML, pero esta se guarda extensión `.php`. Al insertar el código requerido de PHP, este puede ir en cualquier parte del documento, así como dentro de una etiqueta HTML. Para hacer uso de PHP sólo se deben de incluir los símbolos `<?php` para abrir y `?>` para cerrar una vez que haya concluido la operación debida en PHP. Estos símbolos para abrir y cerrar PHP se pueden incluir las veces que sean necesarias en el documento HTML.

En la Figura 8 se muestra cómo está estructurado un script en PHP. Comienza con la etiqueta de HTML `<select>`, la cual ingresa a una lista de opciones los elementos que se requieran, en este caso se usó un ejemplo práctico del sistema, que es para listar la herramienta que se comprará. Después se abre PHP con el símbolo de apertura (`<?php`), se hace la consulta a la tabla herramientas en la base de datos y se almacena en la variable `$re`.

## Script PHP en documento HTML

```
<select id="ddlHerramientas" class="form-control btn btn-info dropdown-
toggle">
<?php
  $re=$conexion->query("SELECT * FROM herramientas")or die(mysqli_error());
  while ($f=mysqli_fetch_array($re)) {
?>
<option value="<?php echo $f['IdHerramienta'];?>"><?php echo $f['Nombre'];
?></option>

<?php
  }
?>

</select>
```

Figura 8. Script PHP en documento HTML.

Posteriormente se inicia un ciclo while para agregar los elementos consultados a la base de datos y dejarlos en la caja de texto en lista. Como se puede observar, después se cierra temporalmente PHP con el símbolo de cierre (?>) para volver a HTML con la etiqueta <option> los cuales son los elementos de la lista; dentro de esta etiqueta <option> se vuelve a abrir PHP para poner el valor y el texto (que será visible para el usuario). A continuación, se ven los símbolos de apertura y cierre de PHP y dentro de ella, se encuentra la llave de cierre del ciclo while. Finaliza con el cierre de la etiqueta <select> en HTML. Ver el resultado en la Figura 9.

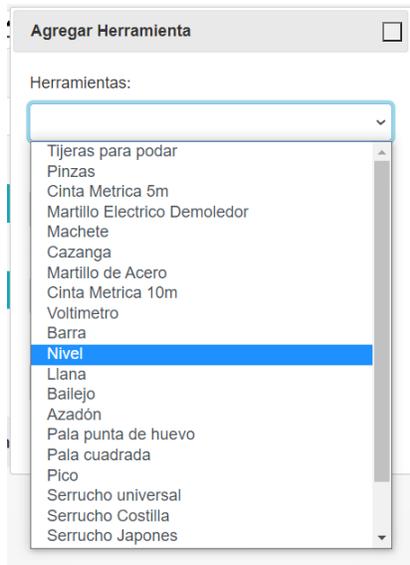


Figura 9. Resultado del script php en documento html.

### 6.4.3 Variables

Existen variables que son símbolos portadores de un valor. Se les llama variables porque su valor no está determinado desde un inicio, sino que hay que asignarlo cuando lo defina el programador desde el código, y éste valor no es único ni permanente, sino que en el transcurso de la ejecución del programa puede ir cambiando si así se requiere. Las variables se nombran libremente, pero en sus nombres no pueden ir palabras reservadas del lenguaje y estas inician con un símbolo de pesos (\$), seguido del nombre que, como buena práctica, debe iniciar con una letra o guion bajo (\_). Ver en el ejemplo práctico del sistema en la Figura 10, que almacena en una variable llamada *\$result* la cantidad de registros que hay en una tabla en MySQL y le aumenta el valor de 1, para así guardar el registro actual sin ser repetido (Arias, 2017).

Variables en PHP

```

<?php
$result=$conexion->query("SELECT count(*) as contador from compras");
$dato=mysqli_fetch_assoc($re);
$id=$dato['contador']+1;
?>
```

Figura 10. Variables en PHP.

### 6.4.3.1 Características

Para una buena práctica de programación en PHP, las variables que se usen deben de tener las siguientes características. (Gómez, 2005).

- No es necesario declarar las variables desde un inicio, basta con declararlas al momento de necesitarlas.
- Las variables no tienen valores predefinidos, puede asignársele cualquier tipo de dato.
- Los nombres de variables comienzan por el signo de pesos (\$).
- Una misma variable se puede reutilizar con un tipo de dato diferente al anterior.
- En los nombres de variables influyen las mayúsculas y minúsculas, de tal manera que \$VARIABLE sería diferente de \$variable.
- Para asignar un valor a la variable basta con usar el signo de igual (=), seguido del valor: si es numérico se pone el número, pero si es de tipo cadena, se tienen que usar las comillas (“ejemplo”) encerrando el valor.

## 6.5 MySQL

Las bases de datos son colecciones de datos estructurados que se relacionan para crear un sentido y dar más eficiencia. Son lugares donde los datos son almacenados y organizados de manera eficiente. Las bases de datos son muy importantes para las empresas, ya que se han convertido en la parte principal de los sistemas de información.

Las bases de datos son gestionadas por un Sistema Gestor de Base de Datos (conocido como SGBD). Su objetivo principal es evitar la manipulación directa por un usuario de una base de datos y establecer un estándar, teniendo un interfaz para que los programas puedan acceder. En la actualidad hay varios SGBD en el mercado, algunos de ellos son: *MySQL*, *Oracle*, *SQL Server*, *PostgreSQL* (Arias, 2014).

### 6.5.1 Queries en MySQL

Los queries son consultas (la palabra búsqueda en inglés es *Query*) u operaciones como eliminación, actualización, etc. que se realizan a la base de datos en MySQL mediante comandos en el motor de búsqueda.

#### 6.5.1.1 Consulta

En la Figura 11 se muestra una consulta hecha en MySQL. La cual extrae los usuarios registrados en el sistema y los muestra si se ejecuta satisfactoriamente el siguiente Query:

```
SELECT IdUsuario, Nombre_Usuario, Telefono_Usuario, Email, Foto_Usuario, Domicilio_Usuario FROM usuarios
```

| IdUsuario | Nombre_Usuario                | Telefono_Usuario | Email                | Foto_Usuario                                | Domicilio_Usuario   |
|-----------|-------------------------------|------------------|----------------------|---|---------------------|
| 1         | Pedro Jaramillo Ventura Rolón | 3411231234       | pjvr_123@hotmail.com | resource/img/users/default.jpg              | Pedro Moreno #32    |
| 2         | María Fernanda López Pérez    | 3417897894       | mflp_123@gmail.com   | resource/img/users/default.jpg              | Avila Camacho #13Q  |
| 3         | José Cortez Larios            | 3414564561       | jcl_123@yahoo.com    | resource/img/users/default.jpg              | Ramón Corona #45    |
| 4         | Miguel Rubio Moreno           | 3416736731       | mrm_123@outlook.es   | resource/img/users/default.jpg              | López Cotilla #154  |
| 5         | Yael Alejandro Santana Michel | 3421006559       | asm_1995@outlook.com | resource/img/users/asm_1995@outlook.com.jpg | Ramón Corona #166-B |
| 6         | Pedro Páramo Pérez            | 3414004004       | ppp@hotmail.com      | resource/img/users/default.jpg              | Pedro Moreno #123   |

Figura 11. Resultado del Query.

### 6.5.1.2 Actualización de registro

En la Figura 12 se muestra la sintaxis de un query de actualización. Como resultado se tendrá una actualización del inventario de herramientas, porque se incrementa la existencia al ser usado en el movimiento de compras. N es el número que se incrementará y IdDefinido el Id de la herramienta comprada.

Actualización de herramientas

```
UPDATE `herramientas` SET Existencia=Existencia+N WHERE `herramientas`.`IdHerramienta` = IdDefinido
```

Figura 12. Query de actualización de la tabla herramientas.

### 6.5.2 Acceso a base de datos desde PHP

Para acceder fácilmente a la base de datos en MySQL se pueden usar diferentes funciones existentes. En la Figura 13 se muestra un ejemplo práctico para dar acceso a la base de datos del sistema desarrollado a través de PHP.

```
<?php
    $host = "localhost";
    $usuario = "root";
    $pass = "";
    $db = "serviWeb";
    $conexion = mysqli_connect($host,$usuario,$pass,$db);
    mysqli_query($conexion,"SET NAMES 'utf8'");
?>
```

Figura 13. Conexión a base de datos.

## 6.6 JSON

JSON son siglas en inglés que son por JavaScript Object Notation. Se trata de un formato para guardar e intercambiar información que cualquier persona puede leer. Los archivos JSON contienen solo texto y usan la extensión *.JSON*. Estos archivos permiten representar un objeto con texto plano. Útil para inicializar objetos o recibirlos desde el servidor (Brent, 2014).

JSON también se define como un formato para almacenar información estructurada y se utiliza principalmente para transferencia de datos entre cliente y servidor. Esto es una alternativa más simple y liviana que XML (lenguaje de marcado extenso, por sus siglas en inglés), el cual tiene funciones similares a JSON.

### 6.6.1 Sintaxis JSON

En JSON existen dos elementos centrales en un objeto: claves (keys) y valores (values). Las claves deben ser cadenas de caracteres (strings) y estas deben de contener una secuencia de caracteres rodeados de comillas. Ejemplo: *“claveEjemplo”*. Los valores son un tipo de dato en JSON. Estos valores pueden ser arreglos, objetos, cadenas, booleanos o nulos.

#### 6.6.1.1 Objeto

Un objeto JSON debe comenzar y terminar con llaves (*{,}*). Puede tener dos o más pares de llaves con datos, separándolos por una coma. Asimismo, cada clave es seguida de dos puntos para distinguirla del valor que contiene. Ver el ejemplo en la Figura 14.

## Objeto JSON

```
“empleado”: { “nombre”: “José Velasco”, “puesto”: “Reparador”,  
“estado”: “Activo” }
```

Figura 14. Objeto JSON.

### 6.6.1.2 Arreglo

Son colecciones ordenadas de valores. Debe comenzar y terminar con corchetes (*[, ]*) y cada valor debe estar separado por una coma. Un arreglo JSON puede contener varios objetos JSON. Ver el ejemplo en la Figura 15, que en este caso es un arreglo que contiene tres objetos.

## Arreglo JSON

```
“empleados”: [  
  { “primerObjeto” : “Ricardo Velasco”, “puesto” : “Reparador”, “estado” :  
    “Inactivo” },  
  { “primerObjeto” : “Alejandra Ramírez”, “puesto” : “Reparador”, “estado” :  
    “Inactivo” },  
  { “primerObjeto” : “Reyes Luján”, “puesto” : “Reparador”, “estado” :  
    “Activo” },  
]
```

Figura 15. Arreglo JSON.

## 6.7 BootStrap

Bootstrap es un framework de código abierto para desarrollo Web con HTML, CSS y JS que permite un diseño responsivo, es decir, que se adapta a los diferentes tamaños de pantalla. Este es uno de los aspectos más destacados al desarrollar un sitio Web con Bootstrap, donde se puede elegir características, así como personalizar distintos componentes.

Bootstrap es un producto de código abierto de Mark Otto y Jacob Thornton que, cuando se lanzó inicialmente, ambos eran empleados de Twitter. Era necesario estandarizar los conjuntos de herramientas FRONTEND de ingenieros de toda la empresa. Desde que Bootstrap fue lanzado en agosto de 2011, ha despegado su popularidad. Ha evolucionado

desde ser un proyecto completamente en CSS hasta incluir gran cantidad de complementos e iconos JavaScript que van de la mano con formularios y botones. (Spurlock, 2013).

### 6.7.1 Características

Ofrece una serie de plantillas CSS y ficheros de JavaScript que permiten integrar el framework de manera sencilla a los proyectos Web. Algunas de sus características principales son:

- Permite crear sitios Web responsivos, es decir, que se adaptan a los diferentes tamaños de pantalla que hay en el mercado a distintas escalas y resoluciones.
- Framework amigable con otras librerías de JavaScript como JQuery.
- Funciona para todos los navegadores.
- Dispone de estructuras fijas para ayudar en el diseño del sitio Web.
- Usa estándares recientes como CSS3 y HTML5.
- Biblioteca de código front-end.

### 6.7.2 Implementación en HTML

Para usar Bootstrap en HTML se debe descargar las hojas de estilo desde la página Web oficial, después se hace la importación como cualquier hoja de estilo en CSS. A continuación, se presenta un ejemplo sencillo en la Figura 16.

```
Implementación en HTML.  
  
<html>  
  <head>  
    <title>Implementación de bootstrap</title>  
    <link href="js/bootstrap.min.css" rel="stylesheet">  
  </head>  
  <body>  
    <h1>¡Hola mundo!</h1>  
    <script src="js/bootstrap.min.js">¡Este es un ejemplo!</script>  
  </body>  
</html>
```

Figura 16. Implementación de bootstrap en HTML.

### 6.7.3 Componentes

Bootstrap ofrece distintos componentes que se pueden usar con unos estilos predefinidos y configurables.

#### 6.7.3.1 Grid básica

La grid (cuadrícula en español) es uno de los componentes principales en Bootstrap, ya que se usa para darle estructura a la página Web. La grid consta de 12 columnas y 940 píxeles de ancho. Ver en la Figura 17 ejemplo de estructuración.

|    |   |   |   |   |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|---|---|---|---|
| 1  | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2  |   | 2 |   | 2 |   | 2 |   | 2 |   | 2 |   |
| 3  |   |   | 3 |   |   | 3 |   |   | 3 |   |   |
| 4  |   |   |   | 4 |   |   |   | 4 |   |   |   |
| 6  |   |   |   |   |   | 6 |   |   |   |   |   |
| 12 |   |   |   |   |   |   |   |   |   |   |   |

Figura 17. Grid con ejemplo de estructuración.

#### 6.7.3.2 Barra de navegación

Para poder hacer una barra de navegación para la página Web se hace uso de la clase *nav-item dropdown*. Con esto se le indica a HTML, que cuando se haga click sobre algún elemento definido en la barra de navegación, se despliegue una lista con lo que contiene en forma de dropdown list.

Con la clase “dropdown-menu” se agregan los elementos a la lista mediante la etiqueta `<a>`. El código se puede observar en la Figura 18.

```

Barra de navegación.

<li class="nav-item dropdown">
  <a class="navlink dropdowntoggle" href="#" id="navbarAdminCatalogos"
      data-toggle="dropdown">
    <span class="fas fa-book"></span> Catálogos
  </a>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#" data-source="empleados.php">
      <span class="fas fa-users">Empleados </span>
    </a>
    <a class="dropdown-item" href="#" data-
source="herramientas.php">
      <span class="fas fa-wrench"> Herramientas </span>
    </a>
    <a class="dropdown-item" href="#" data-source="usuarios.php">
      <span class='fas fa-user-tie'> Usuarios </span>
    </a>
    <a class="dropdown-item" href="#" data-
source="Proveedores.php">
      <span class='fas fa-user-tie'> Proveedores </span>

```

Figura 18. Código de una barra de navegación usando bootstrap.

Resultado:

Una vez que se haya ejecutado el código de la Figura 18 en algún navegador, se podrá observar el resultado como se muestra en la Figura 19.

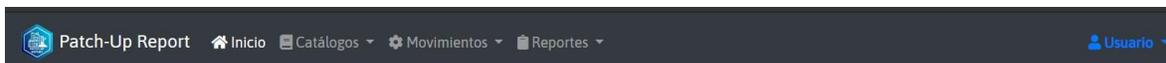


Figura 19. Ejecución de barra de navegación.

### 6.7.3.3 Cajas de texto

Para darles diseño a las cajas de texto con Bootstrap, se deben usar varias clases. Entre ellas “form-group” para dar tamaño y después se define el espacio que ocupará, en este caso será de 6 columnas. Para el input text se hace uso de la clase “form-control”. Algo parecido se hace con todas las cajas de texto que se quieran usar en la página Web, solo se cambia el

tipo de caja de texto y el tamaño que se requiera. En la Figura 20 se puede ver un ejemplo del código.

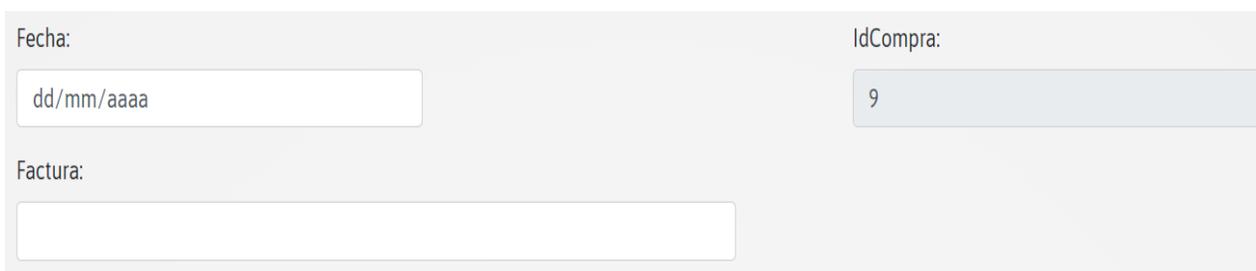
```
Código para implementar Bootstrap en un dropdown

<div class="form-group col-md-6">
  <label for="ddlProveedor">Proveedor:</label>
  <select name="ddlProveedor" id="ddlProveedor" class="form-
  control btn btn-info dropdown-toggle" required>
    <option value="<?php echo $f['IdProveedor'];?>">
      <?php echo $f['Nombre'];?>
    </option>
  </select>
</div>
```

Figura 20. Código de un input con bootstrap.

Resultado:

Una vez que se ejecute el código de la Figura 20, se visualiza el resultado de la Figura 21.



The screenshot shows a light gray background with three input fields. On the left, there is a label 'Fecha:' above a text input field containing 'dd/mm/aaaa'. Below it is a label 'Factura:' above a larger, empty text input field. On the right, there is a label 'IdCompra:' above a text input field containing the number '9'.

Figura 21. Ejecución de caja de texto.

#### 6.7.3.4 Dropdown (select).

Para usar los dropdowns (listas desplegables) con Bootstrap, se hace uso de algunas clases definidas. Como en todos los demás elementos del formulario como cajas de texto, se hace uso de la clase “form-group” para dar tamaño y estructura. Dentro de la etiqueta <select> (que es el dropdown), se emplea la clase “btn btn-info dropdown-toggle”. El *btn-info* se usa para indicar el color, que en este caso es de color azul. El *dropdown-toggle* es la clase principal, ya que desde allí se indica que es un tipo de dropdown. Ver Figura 22.

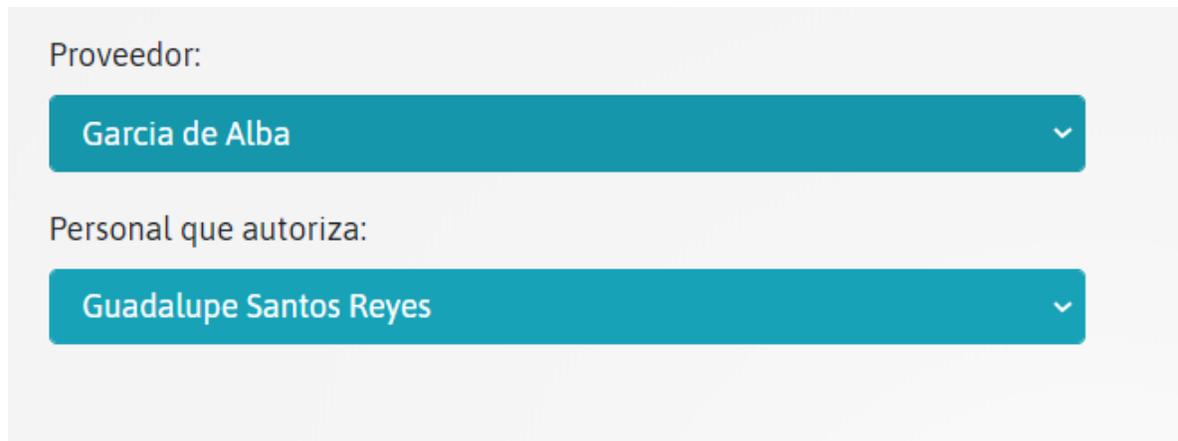
### Código para implementar Bootstrap en un dropdown

```
<div class="form-group col-md-6">
  <label for="ddlProveedor">Proveedor:</label>
  <select name="ddlProveedor" id="ddlProveedor" class="form-
  control btn btn-info dropdown-toggle" required>
    <option value="<?php echo $f['IdProveedor'];?>">
      <?php echo $f['Nombre'];?>
    </option>
  </select>
</div>
```

Figura 22. Código de un dropdown (select) con bootstrap.

Resultado:

Una vez que se ejecute el código de la Figura 22, el resultado se muestra en la Figura 23.



Proveedor:

Garcia de Alba

Personal que autoriza:

Guadalupe Santos Reyes

Figura 23. Ejecución de dropdown.

### 6.7.3.5 Botones

Para el diseño de los botones con Bootstrap, se utiliza “form-group” para establecer un tamaño. Dentro del botón se emplea la clase “btn btn-success btn-xs” esto indica que se requiere el diseño de un botón, seguido del color. En este caso es un *btn-success*, aunque esto puede cambiar según se requiera. Ejemplo del código en la Figura 24.

```
Código para implementar Bootstrap en un botón

<div align="right" class="form-group col-md-2">
  <button type="button" name="add" id="add" class="btn btn-
    success btn-xs"> Agregar Herramienta </button>
</div>
```

Figura 24. Código de un botón con bootstrap.

Resultado:

Una vez que se ejecute el código de la Figura 24, se muestra el resultado en la Figura 25.

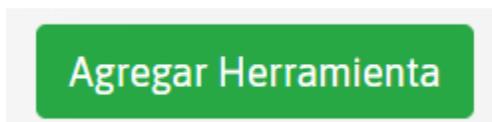


Figura 25. Ejecución de botón.

### 6.7.3.6 Ventana modal

Para hacer uso de una ventana modal sólo se hace uso de la clase *modal* predefinida por Bootstrap, dentro de la etiqueta <div>. Dentro de ella se colocan todos los elementos que se requieran. Un pequeño ejemplo del código principal se muestra en la Figura 26.

```
Ventana modal con Bootstrap.

<div id="user_dialog" title="Agregar Herramienta" class="modal">
  ...
</div>
```

Figura 26. Código principal de una ventana modal.

Resultado:

Una vez que se haya ejecutado el código de la Figura 26 y dentro de las etiquetas *div*, en el modal se hayan agregado los elementos necesarios, se genera el siguiente resultado en la Figura 27.

The image shows a web application interface. A modal window titled "Agregar Herramienta" is open in the center. Inside the modal, there is a dropdown menu labeled "Herramientas" with a teal background. Below it are four input fields: "IdHerramienta", "Cantidad", and "Costo". At the bottom of the modal is a teal "Guardar" button. The background shows a table with columns for "Descripción", "Cant", "Costo", and "Total". To the right of the table, there is a sidebar with buttons for "Agregar" (teal), "Editar" (grey), "Subtotal:" (with a value of "0"), and "Total:".

Figura 27. Ejecución de ventana modal.

## 6.8 Java

Java es un poderoso lenguaje de programación de propósito general capaz de ser ejecutado en múltiples plataformas. Fue creado por la compañía *Sun Microsystems*, pero después Oracle adquirió esta compañía y el lenguaje pasó a ser de su propiedad.

Java es multiplataforma, es decir, que es capaz de ejecutarse en gran parte de los sistemas operativos con una única base de código. Esto se hace posible gracias a una máquina virtual que existe en cada sistema, la cual es capaz de ejecutar Java y de ser una intermediaria entre el lenguaje de programación y el dispositivo. Lo anterior supone una ventaja para los desarrolladores de software, ya que no tienen que adaptar lo programado para cada sistema operativo como Windows, Ubuntu, MacOS, etcétera (Deitel, 2003).

### 6.8.1 Funcionamiento

Para que Java funcione se necesita de un código fuente (el cual es creado por el usuario), este archivo debe tener una extensión `.java`. Una vez que se hayan puesto en el código fuente las instrucciones que se requieren, el compilador toma el archivo con extensión `.java` y lo transforma en Java *byte-code* dentro de un archivo con extensión `.class`. Cuando se quiera ejecutar el programa creado en Java, la máquina virtual leerá el archivo `.class`, el cual le dará las instrucciones a la computadora del sistema operativo que se requiera. Ejemplo del funcionamiento de Java en la Figura 28.

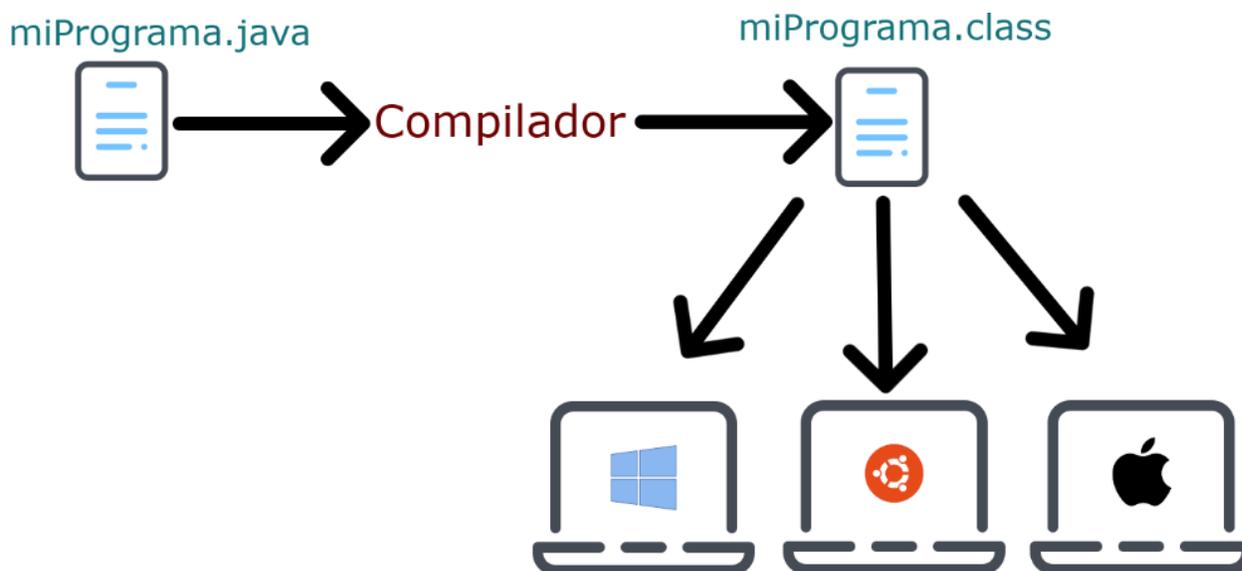


Figura 28. Funcionamiento de Java.

### 6.8.2 Java para Android

Java es uno de los dos lenguajes que se usa para programar aplicaciones nativas para Android, el otro es Kotlin. Java es el lenguaje con el que se le puede sacar mayor aprovechamiento a todas las posibilidades de la plataforma y así, obtener el mejor rendimiento en las aplicaciones desarrolladas para Android.

Esta situación ha permitido que se abra una nueva etapa en el lenguaje de programación Java, ya que en los últimos años han surgido muchos lenguajes de programación que han ido ocupando su lugar. En la Figura 29 se muestra el código fuente de la clase Usuario, la cual es usada en la aplicación para el inicio de sesión de usuarios.



## 6.9 Android Studio

Android Studio es el entorno de desarrollo integrado oficial para la plataforma móvil Android. Este fue anunciado en 2013, pero hasta diciembre de 2014 fue publicada la primera versión estable. Esta reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones Android.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de Apache 2.0. Está disponible en diferentes sistemas operativos, como: Microsoft Windows, MacOS y Linux. Ha sido diseñado específicamente para el desarrollo de Android.

### 6.9.1 Arquitectura

Las aplicaciones Android se ejecutan en un sistema con restricciones (memoria disponible, consumo de batería, visualización, etcétera). Se debe prestar atención a los siguientes puntos:

- Creación de nuevos objetos.
- Uso de recursos (procesador, RAM, memoria interna, etc.).
- Consumo de batería.
- Visualización en los diferentes tamaños de pantalla para que el contenido se presente correctamente.
- Versión de Android.

La arquitectura en Android se compone de cinco partes:

1. **Aplicación:** Representa el conjunto de aplicaciones proporcionadas con Android.
2. **Framework:** Esto permite a los desarrolladores crear aplicaciones accediendo al conjunto de API y funcionalidades disponibles en el teléfono.
3. **Librerías:** Android dispone de un conjunto de librerías que utilizan los distintos elementos de la aplicación.
4. **Runtime:** Es el que contiene, por ejemplo, la máquina virtual.
5. **Kernel:** Es el que proporciona una interfaz con el hardware que gestiona la memoria, los recursos y los procesos.

### 6.9.2 Vistas (Views)

Las vistas son los elementos que componen la interfaz del usuario de una aplicación. Por ejemplo: botones, cajas de texto, imágenes, etcétera. Las vistas se definen mediante un fichero XML, y el sistema crea los objetos a partir del archivo. Esta forma de trabajar es muy similar a la Web con HTML. En la Figura 31 se muestra un fragmento de código con XML que representa un botón en el menú de inicio.

## Código XML

```
<LinearLayout
  android:layout_width="match_parent"
  android:layout_height="match_parent">

  <ImageView
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:src="@drawable/consultagraal" />

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:text="@string/btconsgral"
    android:textColor="@color/primariotres"
    android:textSize="16sp" />

</LinearLayout>
```

Figura 31. Código XML de una vista.

En la Figura 32 se presenta la ejecución del código.

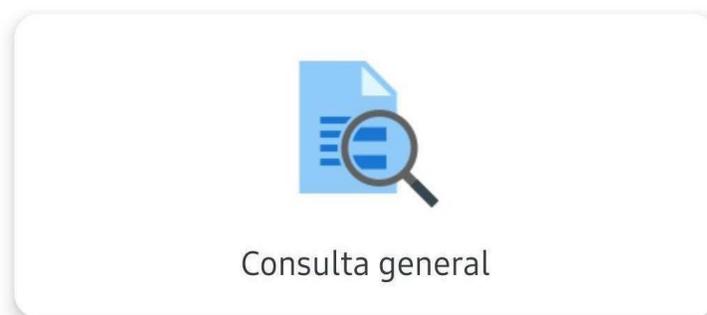


Figura 32. Ejecución del código XML en Android.

### 6.9.3 Layout

Un Layout es un conjunto de vistas agrupadas en una forma determinada. Android da la opción de disponer de diferentes tipos de Layouts, según se quieran agrupar los objetos, ya sea en cuadrícula, de forma lineal, o indicando la posición absoluta de cada vista. Al igual que las vistas, los Layouts pueden ser definidos desde código HTML. Ver un ejemplo en la Figura 33 de un *ConstraintLayout* (Gironés, 2012).

```
Código XML

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://s
chemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/fondoapp"
    tools:context=".MenuActivity">

    <!-- Vistas que se requieran aquí !-->

    ...

    <!--Las vistas se presentarán en el ConstraintLayout !-->

</androidx.constraintlayout.widget.ConstraintLayout>
```

Figura 33. Código XML de Layout.

### 6.9.4 Preferencias (Shared Preferences)

La forma más sencilla de almacenar datos en las aplicaciones es a través de la clase *SharedPreferences*, ésta permite guardar y consultar datos en una forma de clave y valor. De tal manera que la clave será el nombre que se le da al dato que se quiere almacenar y el valor podrá ser de tipo: booleano, flotante, entero, long, cadena, etcétera.

Las preferencias no son más que datos que una aplicación puede guardar para mejorar la experiencia del usuario. Ver el ejemplo en la Figura 34, este código es usado para almacenar el correo y contraseña del usuario.

```

correo = etEmail.getText().toString();
contra = etContra.getText().toString();
if (chkRecuerdame.isChecked()){
    loginPrefsEditor.putBoolean("guardarIngreso", true);
    loginPrefsEditor.putString("correo", correo);
    loginPrefsEditor.putString("contra", contra);
    loginPrefsEditor.commit();
}

```

Figura 34. Guardando correo y contraseña con Preferencias.

### 6.9.5 Intención (Intent)

Una intención representa la actividad de lanzar una opción. Estas intenciones son usadas cada que se quiere hacer una de las siguientes cosas:

- Lanzar una activity.
- Lanzar un servicio.
- Lanzar un anuncio de tipo *broadcast*.
- Comunicarnos con un servicio (como cargar página Web, hacer una llamada, enviar un correo electrónico, etcétera).

Los componentes lanzados pueden ser internos o externos a la aplicación. También se utilizan las intenciones para el intercambio de información entre diferentes componentes.

Para realizar el lanzamiento de una actividad, sólo se hace uso de un objeto de la clase *Intent*. Dos ejemplos de lanzamientos de intentos, uno interno y otro externo. Ver el primero en la Figura 35, el cual abre una Activity dentro de la misma aplicación.

#### Abrir activity dentro de la aplicación

```

cvRegistrarReporte.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        botones(false);
        Intent intent = new Intent(MenuActivity.this, RegReporteActivity.class);
        startActivity(intent);
    }
});

```

Figura 35. Intento interno para abrir activity.

Para el ejemplo externo, en la Figura 36 se ve un ejemplo, el cual se usa para enviar un correo a través de la app de Email que tenga el usuario asignada (Gironés, 2012).

```
Intento para enviar correo a través de la app de email.

public void enviarMail(){
    Intent emailIntent = new Intent(android.content.Intent.ACTION_SEND);
    String aEmailList[] = { "ayuda@davidvelasco.com.mx" };

    emailIntent.setType("plain/text");
    emailIntent.putExtra(android.content.Intent.EXTRA_EMAIL, aEmailList);
    emailIntent.putExtra(android.content.Intent.EXTRA_SUBJECT, "CONTACTO A
PP: " + Usuario.getId());
    emailIntent.putExtra(android.content.Intent.EXTRA_TEXT, Usuario.getNom
bre() );

    startActivity(Intent.createChooser(emailIntent, getString(R.string.env
iar_correo)));
}
```

Figura 36. Intento externo para enviar correo.

### 6.9.6 Permisos

Una aplicación básica de Android no tiene muchos permisos asociados de manera predeterminada. Esto significa que no puede hacer nada que afecte la experiencia del usuario o los datos del dispositivo. Para hacer uso de esos datos (como contactos, mensajes, llamadas, etc.) o funciones (como internet, ubicación, etc.) protegidas del dispositivo, se deben de declarar qué elemntos se van a usar en el manifiesto de la aplicación (androidmanifest.xml). Ver Figura 37.

```
Permisos en el manifiesto de la aplicación.

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-
permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-
permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Figura 37. Permisos en el archivo android manifest.

### 6.9.6.1 Niveles de permisos

Existen dos niveles de permisos dentro del desarrollo de Android Studio; los normales y los riesgosos.

### 6.9.6.2 Permisos normales

En el manifiesto de la aplicación se incluyen permisos normales, es decir, que no representan un gran riesgo para la experiencia del usuario, sus datos y el funcionamiento del dispositivo. Algunos ejemplos de permisos normales son: Permiso de Internet, permiso para establecer la zona horaria, consumo de batería, establecer alarma, leer ajustes de sincronización, etc.

### 6.9.6.3 Permisos peligrosos

En el manifiesto de la aplicación se encuentran permisos peligrosos, es decir, que representen una vulnerabilidad para los datos del usuario, la experiencia o el funcionamiento de la aplicación. Algunos ejemplos de permisos peligrosos son: permiso de almacenamiento, ubicación detallada y no detallada (*Coarse y Fine*), teléfono, mensajes de texto, contactos, calendario, cámara, sensores corporales, audio, etcétera.

### 6.9.6.4 Permisos en tiempo de ejecución

A partir del API 23 de Android (6.0, *Marshmallow*), los permisos peligrosos ya no se solicitan al momento de la instalación, ahora los usuarios otorgan permiso mientras la aplicación se está ejecutando. Esto no anula la declaración en el manifiesto de la aplicación, estos deben de ser declarados ahí y, para hacerlos válidos, deben de ser autorizados por el usuario cuando está ejecutando la app. Esto significa un mayor control del usuario sobre su experiencia, ya que, si así lo requiere, puede autorizar algunos y negar otros. Por ejemplo, un usuario en la aplicación desarrollada puede otorgar los permisos de la cámara y almacenamiento y a la vez, negar el de ubicación. Estos permisos son revocables en cualquier momento desde la configuración de la aplicación.

En la Figura 38 se puede ver el código en Java para solicitar permisos peligrosos en tiempo de ejecución.

## Código java para solicitar permisos peligrosos en tiempo de ejecución

```
alertOpciones.setItems(opciones, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        if(opciones[which].equals(getString(R.string.si))){
            Intent intent = new Intent();
            intent.setAction(Settings.ACTION_APPLICATION_DETAILS_SETTINGS)
;
            Uri uri = Uri.fromParts("package", getPackageName(), null);
            intent.setData(uri);
            startActivity(intent);
        }else{
            dialog.dismiss();
            Snackbar snackbar = Snackbar.make(cvRegistrarReporte, getStrin
g(R.string.permisos_no_aceptados), Snackbar.LENGTH_LONG);
            snackbar.show();
        }
    }
});
```

Figura 38. Solicitar permisos peligrosos.

En la Figura 39 se puede ver cómo se le presentan al usuario las solicitudes de permisos, según se hayan declarado.

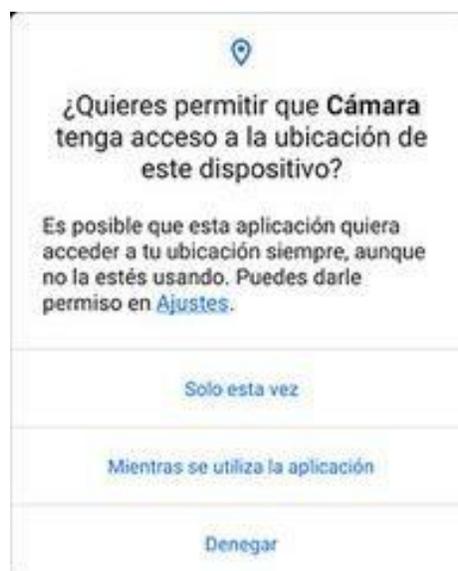


Figura 39. Solicitud al usuario de permiso peligroso.

## 6.9.7 MySQL desde Android Studio

Para la programación de la aplicación móvil elaborada en la presente tesis, se usó la arquitectura *REST*. La cual consume servicios Web desde Android Studio y se explica a continuación.

El acceso a MySQL se hace mediante servicios Web. Es decir, desde Android Studio se consulta un documento PHP, el cual tiene configuración para acceder a la base de datos; se realizan las operaciones y, al final, se imprime el resultado en un formato JSON. Este es el que es rescatado por Android Studio y puede ser manipulado para mostrarlo visualmente al usuario si así lo requiere. Ver en la Figura 40 el código para realizar una consulta a la base de datos e imprimirla en formato JSON. En este caso es un ejemplo práctico del sistema, ya que éste código se usa para hacer la consulta de los resultados de un reporte

```
Consulta e impresión de resultados en formato JSON.

<?php
    $idRep=$_GET['idRep'];
    $consulta="SELECT re.Observaciones, re.Fecha AS FechaTerminacion, dre.Imagen FROM reporteempleado AS re INNER JOIN detallereporteempleado AS dre ON re.IdReporteEmpleado=dre.IdReporteEmpleado INNER JOIN trabajos AS t ON re.IdTrabajo=t.IdTrabajo WHERE t.IdReporteUsuario=$idRep";
    $resultado=mysqli_query($conexion,$consulta);
    if($registro=mysqli_fetch_array($resultado)){
        $result["Observaciones"]=$registro[0];
        $result["FechaTerminacion"]=$registro[1];
        $JSON['reporteusuario'][]=$result;
    }
    echo JSON_encode($JSON);
    mysqli_close($conexion);
?>
```

Figura 40. Acceso a base de datos desde Android Studio.

## 6.10 MP AndroidChart

MP Android Chart es una biblioteca de gráficos de Android que admite 8 tipos diferentes de gráficos, así como gráficos combinados, gráficos de líneas, barras, pastel. La biblioteca ofrece muchas funciones de personalización para adaptarse a las necesidades del usuario. La biblioteca está desarrollada por Philipp Jahoda y se mejora constantemente, enriqueciéndose con cada vez más funciones (PhilJay, 2020).

En la Figura 41, se puede ver su uso en XML para una gráfica de barras.

Gráfica de barras en XML.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <com.github.mikephil.charting.charts.BarChart
        android:id="@+id/barchart"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```

Figura 41. Uso de gráfica de barras con la biblioteca Mp AndroidChart.

## 6.11 Servicios de Google (Play Services)

Los servicios de Google Play es una librería creada por Google, para que los desarrolladores puedan aprovechar las API y servicios que ofrece Google fácilmente.

### 6.11.1 Mapas

Con los servicios de Mapas, se pueden agregar mapas basados en datos de Google Maps a una aplicación. La API controla la descarga de datos, la visualización de mapas y las interacciones de gestos con el mapa. Con la misma API se pueden agregar marcadores, polígonos, gestionar la apariencia del mapa, etc.

Para crear la API que se utiliza en la aplicación, solo hay que dirigirse a la página oficial de *Google Cloud Platform*, una vez ahí, se crea un nuevo proyecto y dentro de esa opción se crea una nueva credencial, la cual será utilizada en la aplicación Android. Ver ejemplo en la Figura 42.

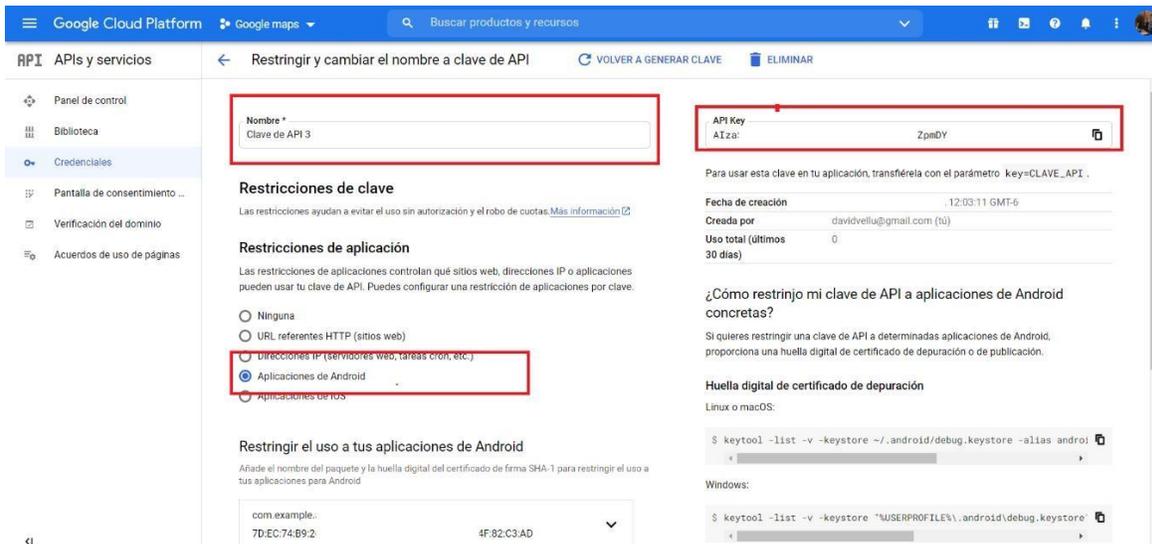


Figura 42. Creación de credencial para uso de mapas de google.

Una vez generada la API Key, como se ve en el cuadro con borde rojo de la esquina superior derecha, se copia y se coloca como un recurso dentro de las etiquetas String en el archivo google\_maps\_key.xml. Se aprecia un ejemplo del código en la Figura 43.

Archivo que contiene el API de los mapas.

```

<resources>
  <string name="google_maps_key" templateMergeStrategy="preserve" translatable="false">YOUR_KEY_HERE</string>
</resources>

```

Figura 43. Archivo con el API de mapas.

### 6.11.2 Localización

Al principio, el uso del GPS era restringido para el uso militar por el departamento de Defensa de Estados Unidos. En la década de 1980 se liberó para el uso civil, aunque su funcionamiento y la exactitud era limitado. Estos servicios sirven para localizar al usuario y dar una mejor experiencia dentro de la aplicación, por ejemplo, en una aplicación de reporte de daños a infraestructura, como esta, se usa la ubicación para poner un marcador en la posición del usuario y así, facilitar el reporte (Lee, 2020).

Ejemplo del uso de ubicación del usuario para agregar un marcador en la Figura 44.

## 6.12 RecyclerView

Uso de Localización para agregar un marcador.

```
private Location loc;
private LocationManager locationManager;

loc = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
if(loc != null){
    latitudGPS = loc.getLatitude();
    longitudGPS = loc.getLongitude();
    mMap.addMarker(new MarkerOptions()
        .position(new LatLng(loc.getLatitude(), loc.getLongitude()))
        .title(getString(R.string.ubicacion_reporte))
        .draggable(true)
        .snippet(getString(R.string.seleccionar))
        .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_MAGENTA)));
    LatLng pos = new LatLng(loc.getLatitude(), loc.getLongitude());
    mMap.moveCamera(CameraUpdateFactory.newLatLng(pos));
    latitud = loc.getLatitude();
    longitud = loc.getLongitude();
}
```

Figura 44. Uso de servicio de localización.

El RecyclerView es un contenedor de elementos; este es sucesor de ListView según los propios desarrolladores de Google. En este modelo, varios componentes diferentes trabajan juntos para mostrar los datos deseados. El objeto RecyclerView se completa con las diferentes vistas que brinda el programador.

### 6.12.1 Uso

Se poder acceder a las funcionalidades de RecyclerView, se agrega la biblioteca al proyecto en el archivo build.gradle del módulo de la aplicación. Esto se hace de la siguiente manera:

```
dependencies {
    implementation 'com.android.support:recyclerview-v7:28.0.0'
}
```

Para usarlo se necesita una fuente de datos, que en este caso provienen de la base de datos, estos se ponen en forma de un Arreglo o Lista, y un adaptador que será el encargado de leer los datos, procesarlos e inflarlos en la lista para enviárselos al RecyclerView y poderlos visualizar. En la Figura 45 se puede ver cómo se define un RecyclerView.

```
Definición de RecyclerView

<android.support.v7.widget.RecyclerView
    android:id="@+id/recycler_view"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />
```

Figura 45. RecyclerView.

Después de que se definió, se crea la vista que estará utilizando el RecyclerView, para mostrarla en una consulta. En este caso se presenta dentro de un CardView. Ver el ejemplo en la Figura 46.

```
Vista de RecyclerView

<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <androidx.cardview.widget.CardView
        android:id="@+id/cardView"
        app:layout_constraintTop_toTopOf="parent">

        <!-- Todos los elementos que se requieran aquí -->

    </androidx.cardview.widget.CardView>

</androidx.constraintlayout.widget.ConstraintLayout>
```

Figura 46. Vista de RecyclerView.

Una vez que se haya completado de codificar tanto la vista como el adaptador y su fuente de datos, que se programa en código Java, la consulta se verá como a continuación se aprecia en la Figura 47. En este caso, como es un sistema multiplataforma, los datos de esta consulta vienen de MySQL, traídos mediante el uso de Servicios Web.



*Figura 47. Visualización de una consulta con RecyclerView.*

## 6.13 Glide

Glide es una librería para Android que fue introducida por google, la cual es usada para descargar y guardar en memoria caché las imágenes que se desean mostrar en la aplicación. Esto lo hace trayendo las imágenes directamente desde el servidor, simplemente proporcionando la URL donde se encuentra alojada la imagen.

Glide tiene la facilidad de cargar la imagen, así como editar el tamaño o la forma en la que se muestra, también se pueden reproducir gif's.

Para instalar la librería de Glide en el proyecto se agrega la biblioteca al proyecto en el archivo build.gradle del módulo de la aplicación. Esto se hace de la siguiente manera:

```
dependencies {  
    implementation 'com.github.bumptech.glide:glide:3.7.0'  
}
```

### 6.13.1 Uso

Para hacer uso de Glide lo primero que se debe tener, es un contenedor para imagen, en este caso se utiliza un CircleView. Ver en la Figura 48.

```
Cargando imagen con Glide desde servidor

String urlImagen;

ivImagen = (CircleImageView)findViewById(R.id.ivImagen);

urlImagen = "http://davidvelasco.com.mx/images/imagenServidor.jpg";
Glide.with(this)
    .load(urlImagen)
    .into(ivImagen);
```

Figura 48. Usando CircleView

Una vez que se tenga el contenedor para la imagen, ahora se hace uso de la librería Glide mediante código Java. Como se puede ver en el ejemplo de la Figura 49, sólo se necesitan dos cosas para cargar la imagen al circleview desde el servidor: la URL donde se encuentra alojada la imagen, el contexto y el lugar donde se mostrará.

```
Contenedor de la imagen

<de.hdodenhof.circleimageview.CircleImageView
    android:id="@+id/ivImagen"
    android:layout_width="100dp"
    android:layout_height="100dp"
    android:src="@drawable/avatar"
    app:civ_border_color="@color/primariotres"
    app:civ_border_width="2dp"/>
```

Figura 49. Uso de Glide.

## 6.14 Material Design

Material Design es una herramienta para el diseño visual, interactivo y de movimiento en los dispositivos. Con este framework se ayuda a dar sombras, tema basado en material de la aplicación, listas, tarjetas, etcétera (Android, 2020).

Para instalar la librería de Material Design en la biblioteca de la aplicación, simplemente hay que dirigirse al archivo `build.gradle` en el módulo de la app. Una vez ahí, se tiene que instalar escribiendo lo siguiente:

```
dependencies {  
    implementation 'com.google.android.material:material:1.0.0'  
}
```

### 6.14.1 Animación de actividades

A partir del API nivel 21 (Android 5.0), se pueden crear transiciones entre las actividades separadas, es decir, de una activity a otra. Para hacer esto hay que hacer uso de `startActivity()`, pero ahora, el programador proporciona un conjunto de opciones mediante `ActivityOptions.makeSceneTransitionAnimation()`.

Para empezar con las transiciones personalizadas, se deben habilitar mediante el atributo `Android:windowActivityTransitions`, esto se hace con código XML desde el archivo de estilos. Ver ejemplo en la Figura 50.

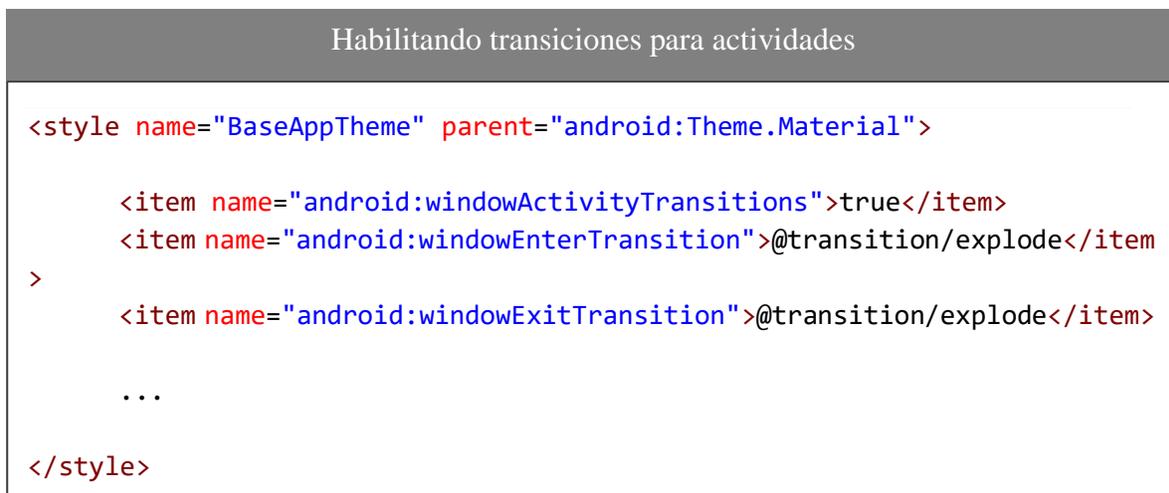


Figura 50. Habilitando transiciones.

Una vez que se ha hecho eso, para hacer uso de las transiciones en java se hace uso del *makeSceneAnimation* y se inicia la actividad. Ver ejemplo en la Figura 51 en el que se inicia otra actividad al presionar un botón.

```
Uso de las transiciones.

View androidRobotView = findViewById(R.id.is);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(this, ReporteActivity.class);
        //Definir la transición
        ActivityOptions options = ActivityOptions
            .makeSceneTransitionAnimation(this, androidRobotView, "robot");
        // Iniciando la actividad normal
        startActivity(intent, options.toBundle());
    }
});
```

Figura 51. Uso de transiciones.

## 6.15 Firebase

Firebase es una plataforma digital que se utiliza para facilitar el desarrollo de aplicaciones Web o móviles de una forma eficiente, rápida y fácil. Es utilizada para, entre otras cosas, medir las métricas de una aplicación, como alcance, descargas, vistas, etcétera. Con eso se puede hacer marketing digital y aumentar la exposición de la aplicación, llegando a usuarios potenciales.

Su principal objetivo es mejorar el rendimiento de las aplicaciones mediante la implementación de diversas funcionalidades que harán la app más manejable, segura y de fácil acceso. (Firebase, 2019)

Para instalar la librería de Firebase en el proyecto primero se debe registrar una cuenta en la página oficial de Firebase (Algo similar a lo que se hace para generar el API de Mapas); posteriormente se agrega la biblioteca al proyecto en el archivo build.gradle del módulo de la aplicación. Esto se hace de la siguiente manera:

```
dependencies {
    implementation 'com.google.firebase:firebase-core:17.0.0'
}
```

### 6.15.1 Google Analytics

Google Analytics recopila datos de uso y comportamiento de la aplicación. Firebase registra dos tipos principales de información.

1. Todos los eventos que suceden en la aplicación, como las acciones de los usuarios, los eventos y los errores.
2. Las preferencias del usuario, tales como las preferencias preferencia de idioma o la ubicación regional.

Para hacer uso de la librería Google Analytics, aparte de Firebase, se debe poner el siguiente código en el proyecto Android:

```
implementation 'com.google.firebase:firebase-analytics:17.2.0'
```

Para registrar eventos predefinidos o personalizados hace falta hacer uso del método `logEvent()`. En el código de la Figura 52 se registra un evento `SELECT_CONTENT` cuando un usuario hace click en un botón de la aplicación.

```
Registro de evento SELECT_CONTENT

Bundle bundle = new Bundle();

bundle.putString(FirebaseAnalytics.Param.ITEM_ID, id);
bundle.putString(FirebaseAnalytics.Param.ITEM_NAME, name);
bundle.putString(FirebaseAnalytics.Param.CONTENT_TYPE, "image");

mFirebaseAnalytics.logEvent(FirebaseAnalytics.Event.SELECT_CONTENT, bundle
);
```

Figura 52. Uso de Google Analytics (firebase).

### 6.16 Volley

Volley es una biblioteca HTTP que facilita y agiliza el uso de redes en aplicaciones Android. Ofrece beneficios como varias conexiones simultáneas, API de cancelación de solicitudes, priorización de solicitudes, herramientas de depuración y rastreo, etcétera.

Esta biblioteca se integra fácilmente con cualquier protocolo de red, además de incluir compatibilidad con cadenas sin procesar, imágenes y JSON. Sólo se recomienda para la transmisión o descarga de archivos pequeños, ya que las respuestas son almacenadas en la memoria durante el análisis.

La biblioteca principal de Volley se desarrolla en GitHub. La manera más fácil de incorporar Volley al proyecto Android, es agregando la siguiente dependencia al archivo `build.gradle` de la aplicación.

```
dependencies {  
    ...  
    implementation 'com.android.volley:volley:1.1.1'  
}
```

Para usar esta librería sin ningún problema, es necesario agregar el permiso `android.permission.INTERNET` en el `androidmanifest.xml`. Sin este permiso, la aplicación no podrá conectarse a la red y esto imposibilitará el funcionamiento de Volley.

### 6.16.1 Consumo servicios Web

Para consumir servicios Web usando Volley, es necesario realizar una solicitud `RequestQueue` y, esta se encargará de ejecutar las operaciones de red, así como leer, escribir la memoria y analizar las respuestas. Las solicitudes analizan las respuestas sin procesar y Volley se encarga de devolver la respuesta analizada al subproceso principal para su entrega.

Para consumir servicios Web utilizando Volley, se debe tener creado un servicio Web y este debe estar alojado en un servidor local o remoto, según se requiera. Ejemplo en la Figura 53 un servicio Web programado en PHP; la salida es en formato JSON y su función en la aplicación es llenar el Spinner con los diferentes tipos de reporte.

```
Servicio Web para llenar Spinner con los tipos de reporte  
  
<?php  
  
require('configuracion/conexion.php');  
$sql = "SELECT Tipo From tiporeporte ";  
$resultado = $conexion -> query($sql);  
  
while($fila=$resultado->fetch_array(MYSQLI_NUM) ){  
    echo JSON_encode($fila);  
}  
$resultado -> close();  
  
?>
```

Figura 53. Creación de servicio Web con PHP.

Si se consulta en la Web lo programado en la Figura 54, se verá el formato JSON con 4 tipos de reportes que existen en la base de datos (Bacheo, áreas verdes, drenaje y agua potable). Ver la Figura 54 donde se presenta la ejecución.



Figura 54. Consulta de servicio Web desde el navegador.

Una vez que se haya creado el archivo PHP con el servicio Web y se encuentre alojado en un servidor local o remoto, podrá ser consumido desde Android Studio con ayuda de la librería Volley. En la Figura 55 se muestra el proceso para llenar el Spinner consumiendo el servicio Web previamente creado. Si se observa hay dos métodos principales: *onResponse* y *onErrorResponse*. El primero se ejecuta cuando Volley analizó y todo está bien; el segundo cuando pasa lo contrario.



Figura 55. Consumo de servicio Web en Android Studio.

## 6.17 JitPack.io

JitPack.io permite construir proyector Git bajo demanda, proporcionando artefactos (jar o aar) listos para usarse. Para usarse con JitPack, el único requisito es que el código del repositorio Git (GitHub, BitBucket, GiLab, etc) que se vaya a usar se encuentre liberado. JitPack facilita el desarrollo, ya que no es necesario clonar el repositorio para acceder a él, como sucedía antes, ahora simplemente se añade la dependencia.

Para comenzar a hacer uso de JitPack.io es necesario añadir la siguiente línea de código de Maven JitPack a la lista de repositorios en el fichero raíz `gradle`.

```
allprojects {
    repositories {
        jcenter()
        maven { url "https://jitpack.io" }
    }
}
```

También se debe añadir la siguiente línea en el archivo `build.gradle` de la aplicación. Con esto ya se podrá empezar a utilizar JitPack.io.

```
dependencies {
    ...
    implementation 'com.github.jitpack:android-example:1.0.1'
}
```

Una vez realizado lo anterior, es posible hacer uso de JitPack. Por ejemplo, para hacer uso de la librería `MPAndroidChart`, simplemente se tiene que añadir la siguiente línea de código en el archivo `build.gradle`

```
dependencies {
    implementation 'com.github.PhilJay:MPAndroidChart:v3.0.3'
}
```

## 6.18 Libphone number

LibPhone Number es una librería de google para aplicaciones móviles y Web, esta sirve para parsear, dar formato y validar un teléfono (sobre todo cuando se trata de uno con lada internacional).

Algunas de las ventajas de esta librería son las siguientes:

- Buscar números en un texto.
- Validar número.
- Ver región del número.
- Da formato al número en tiempo real.
- Proporciona información geográfica del número.
- Proporciona zona horaria relacionada con el número telefónico.

Para instalar esta librería en la biblioteca de la aplicación, hay que añadir la siguiente línea de código en el archivo build.gradle.

```
dependencies {  
    ...  
    implementation 'com.michaelrocks:libphonenumber-android:8.12.8'  
}
```

Se presenta un ejemplo en la Figura 56 del de LibPhone Number uso en Java.

```
Registro de evento SELECT_CONTENT  
  
//Comprobando si el teléfono es válido.  
boolean isValid = phoneUtil.isValidNumber(swissNumberProto);  
  
// Separa la lada del número: Ejemplo  
System.out.println(phoneUtil.format(swissNumberProto, PhoneNumberFormat.INTERNATIONAL));  
// Quita lada internacional al teléfono  
System.out.println(phoneUtil.format(swissNumberProto, PhoneNumberFormat.NATIONAL));
```

Figura 56. Uso de LibPhone en Java.

Es posible comprobar todo lo que la librería ofrece desde la página Web oficial. Ver la Figura 57.

Phone Number entered: +523411143457

defaultCountry entered:

Language entered: en

| Parsing Result (parseAndKeepRawInput()) |                            |
|---|----------------------------|
| country_code                            | 52                         |
| national_number                         | 3411143457                 |
| extension                               |                            |
| country_code_source                     | FROM_NUMBER_WITH_PLUS_SIGN |
| italian_leading_zero                    | false                      |
| raw_input                               | +523411143457              |

| Validation Results             |                      |
|--------------------------------|----------------------|
| Result from isPossibleNumber() | true                 |
| Result from isValidNumber()    | true                 |
| Phone Number region            | MX                   |
| Result from getNumberType()    | FIXED_LINE_OR_MOBILE |

| Formatting Results            |                     |
|-------------------------------|---------------------|
| E164 format                   | +523411143457       |
| Original format               | +52 341 114 3457    |
| National format               | 341 114 3457        |
| International format          | +52 341 114 3457    |
| Out-of-country format from US | 011 52 341 114 3457 |
| Out-of-country format from CH | 00 52 341 114 3457  |

AsYouTypeFormatter Results

Figura 57. LibPhone desde la Web.

## Capítulo 7. Metodología

Para el desarrollo de este proyecto se siguió la metodología de modelo de cascada, el cual consiste en ir por orden de cada una de las etapas del proceso de desarrollo de software. Este modelo consta de 5 fases, las cuales son: análisis de requerimientos, diseño del sistema, codificación, pruebas e implantación.

### 7.1 Análisis de requerimientos

Se realizó un análisis de los requerimientos en la oficina de obras públicas del H. Ayuntamiento después de recabar los requerimientos a través de varias entrevistas con los responsables de obras públicas.

Los requerimientos que se detectaron fueron los siguientes:

- Almacenar información sobre los reportes realizados por usuarios.
- Llevar control de los puntos que demandan mayor atención de mantenimiento.
- Registrar entradas y salidas de herramientas asignadas a tareas de reparación.
- Asignar flotillas de personal para realizar reparaciones de los reportes emitidos, representados por un responsable.
- Ofrecer seguimiento de reportes realizados a través de consultas por los usuarios en una aplicación móvil.

Considerando que los daños en la infraestructura pública son reportados por los propios ciudadanos, se optó por desarrollar una aplicación móvil que permitiera tomar captura de imágenes y enviarlas de manera inmediata a una base de datos en un servidor en la nube, disponiendo además de una aplicación Web para uso exclusivo de los usuarios en la dependencia pública que permitirá atender los reportes en tiempo real y administrar su seguimiento correspondiente. Esto permite incrementar la eficiencia en la atención a la infraestructura reduciendo tiempos, costos y recursos humanos, mejorando además los servicios en la ciudad.

Del anterior análisis de requerimientos se definieron entidades externas, procesos y almacenes de datos, los cuales se van a enlistar a continuación.

#### 7.1.1 Entidades Externas

- Administrador.
- Usuarios.
- Empleados.
- Proveedores.

### **7.1.2 Procesos**

- Trabajos.
- Compras.
- Devoluciones.
- Bajas.

### **7.1.3 Almacenes de datos**

- Proveedores.
- Compras.
- Detalle Compra.
- Baja de herramientas.
- Detalle baja de herramientas.
- Herramientas.
- Tipo herramienta.
- Empleados.
- Devolución de herramientas.
- Detalle devolución de herramientas.
- Usuarios.
- Reporte usuario.
- Detalle reporte usuario.
- Tipo reporte.

## **7.2 Diseño**

A continuación, se verán todos los diagramas usados para el diseño del sistema.

### **7.2.1 Diagrama de caso de uso general**

En la Figura 58 se muestra el diagrama de caso de uso general de la plataforma Web y móvil. En este diagrama se describen las opciones a las que tiene acceso cada una de las entidades externas que interactúan en la plataforma.

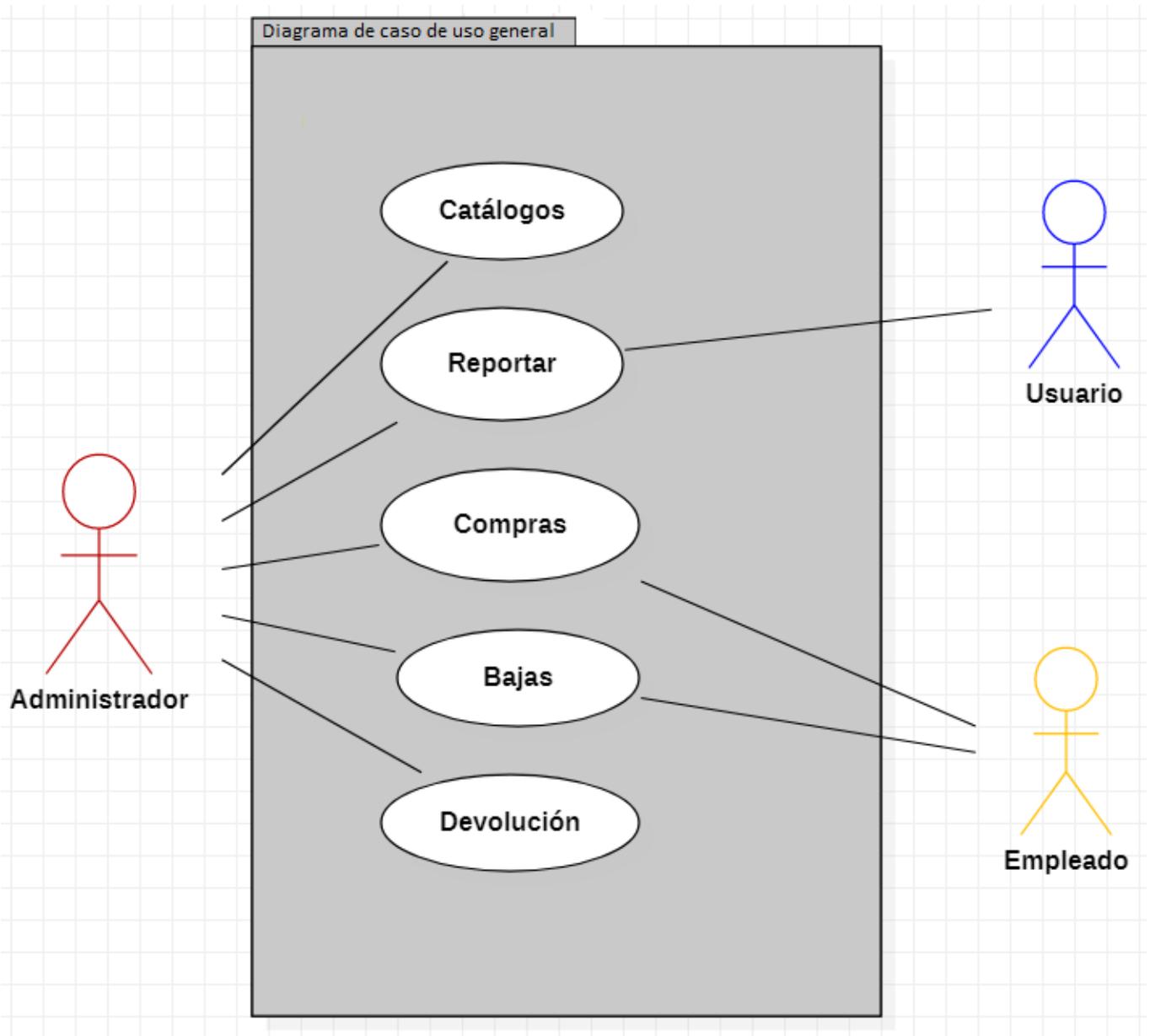


Figura 58. Diagrama de caso de uso general

## 7.2.2 Diagrama de Warnier/Orr para el Administrador

En la Figura 59 se muestra el diagrama de Warnier/Orr, el cual muestra las diferentes opciones y procesos que se llevan a cabo dentro del perfil Administrador.

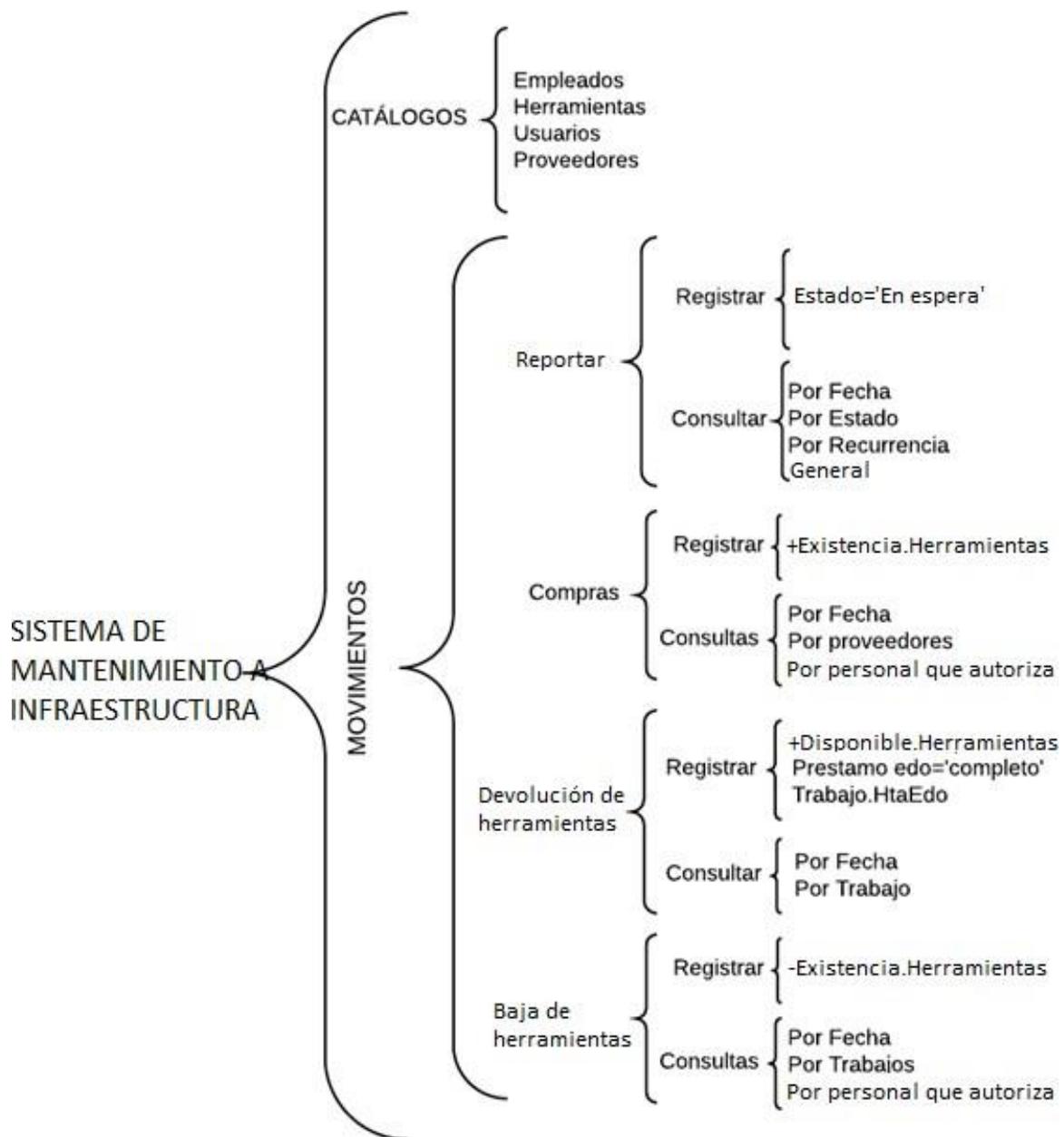


Figura 59. Diagrama de Warnier/Orr para el Administrador

### 7.2.3 Diagrama de Warnier/Orr para el usuario

En la Figura 60 se muestra el diagrama de Warnier/Orr para el usuario.



Figura 60. Diagrama de Warnier/Orr para el Usuario.

### 7.2.4 Diagrama de Warnier/Orr para el empleado

En la Figura 61 se muestra el diagrama de Warnier/Orr para el Empleado.

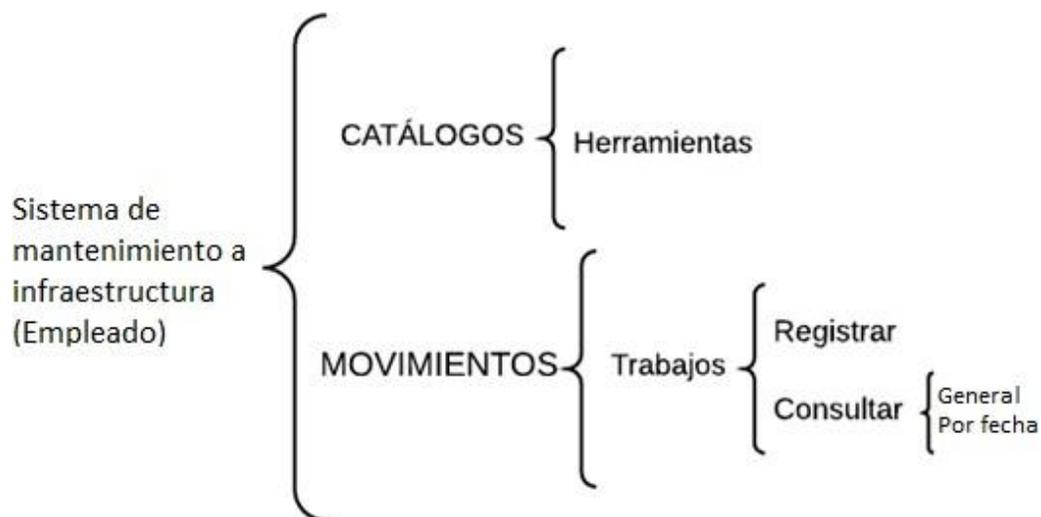


Figura 61. Diagrama de Warnier/Orr para el Empleado.

## 7.2.5 Diagrama de Clases

En la Figura 62 se muestra el diagrama de clases que se diseñó para el desarrollo de la plataforma Web y móvil.

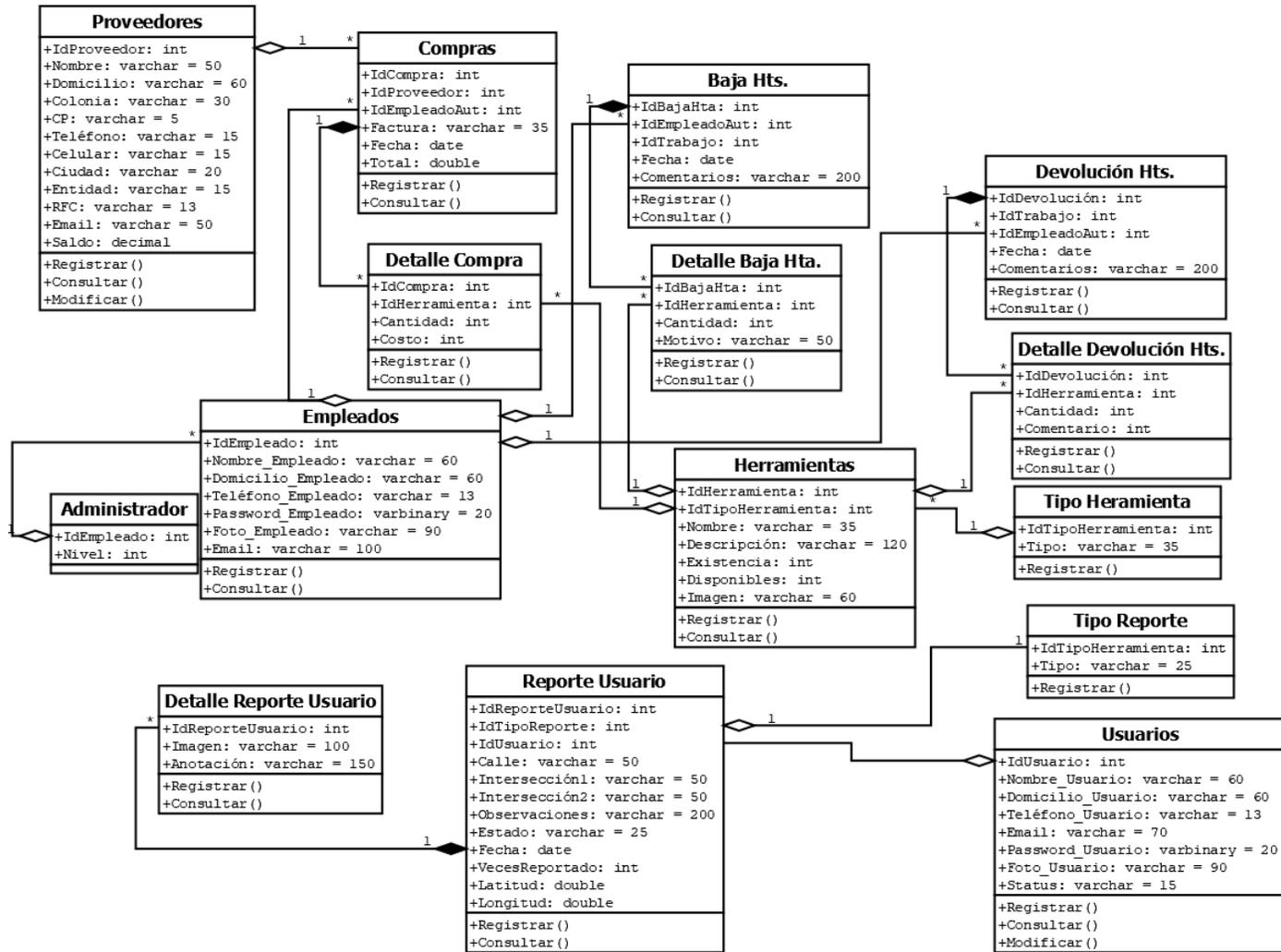


Figura 62. Diagrama de clases

## 7.2.6 Diagramas de Compras

A continuación, se muestran los diagramas utilizados para el movimiento *Compras*, que se encuentra en la Web.

### 7.2.6.1 Diagrama de caso de uso expandido para Registrar Compra

En la Figura 63 se muestra el diagrama de caso de uso expandido para *Registrar Compra*.

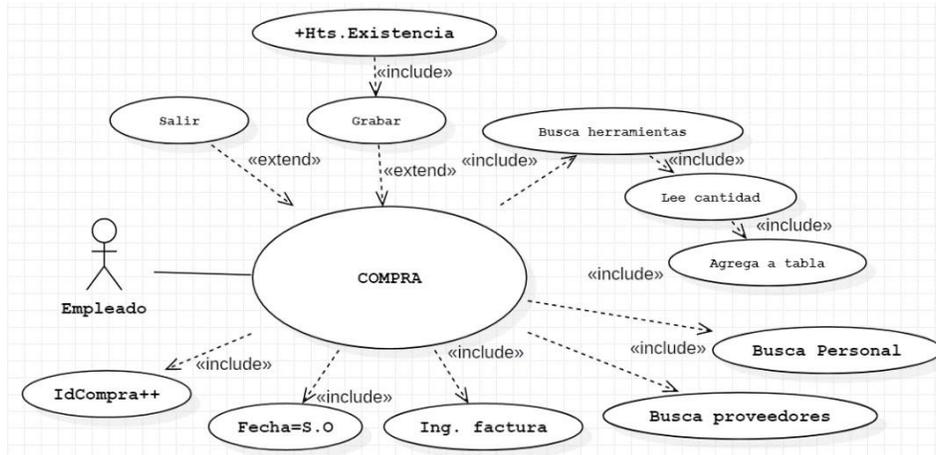


Figura 63. Diagrama de caso de uso expandido para registrar Compra.

### 7.2.6.2 Diagrama de flujo de datos del proceso Registrar Compra

En la Figura 64 se muestra el diagrama de flujo de datos para *Registrar Compra*.

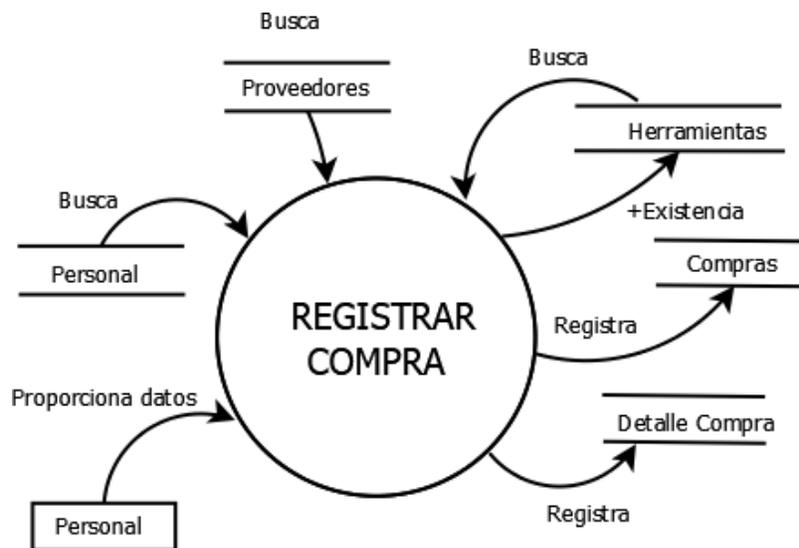


Figura 64. Diagrama de flujo de datos para Registrar Compra.

### 7.2.6.3 Diagrama de colaboración del caso de uso Registrar Compra

En la Figura 65 se muestra el diagrama de colaboración para el caso de uso *Registrar Compra*.

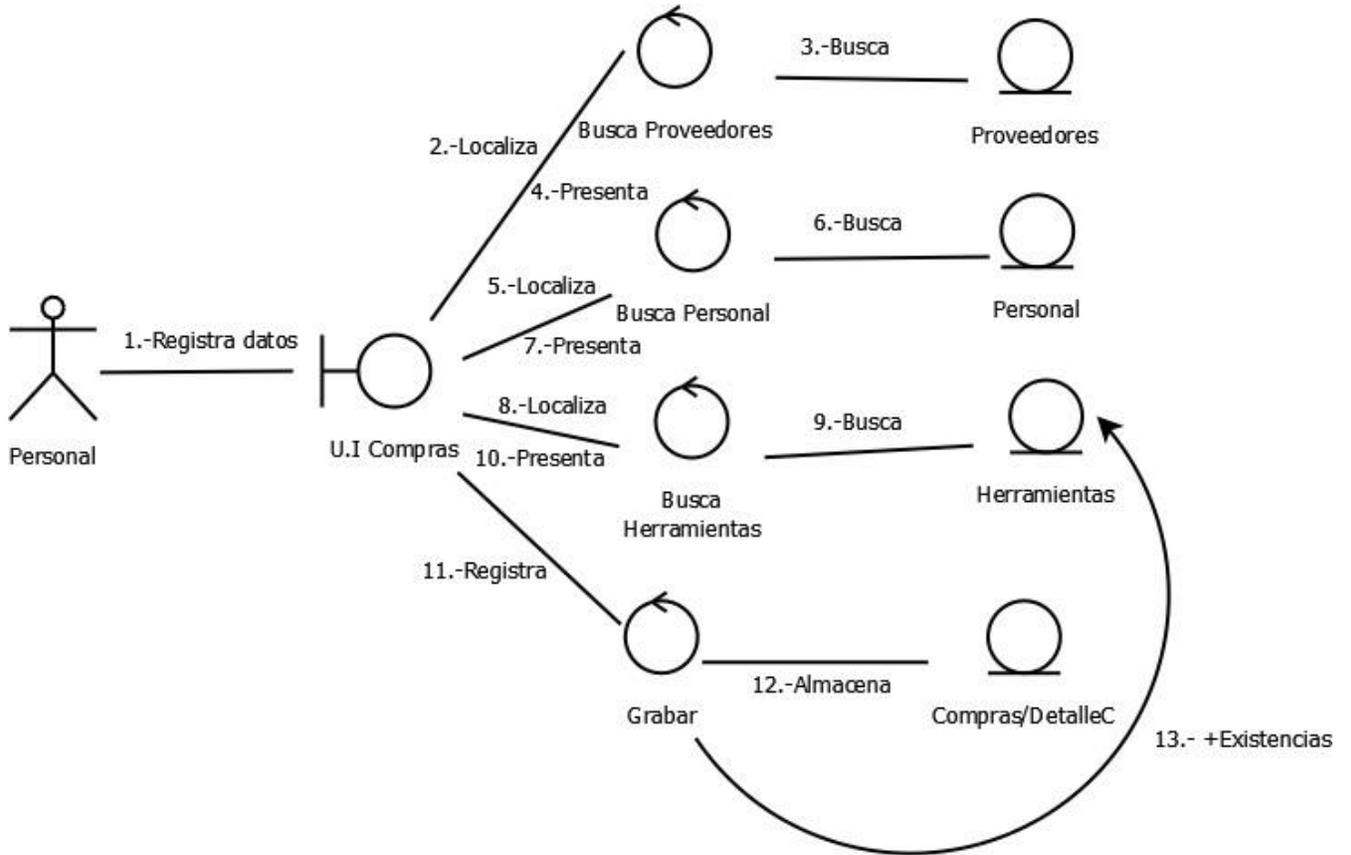


Figura 65. Diagrama de colaboración para el caso de uso Registrar Compra.

#### 7.2.6.4 Diagrama de estados para el caso de uso Compras

En la Figura 66 se muestra el diagrama de estados para el caso de uso *Registrar Compras*.

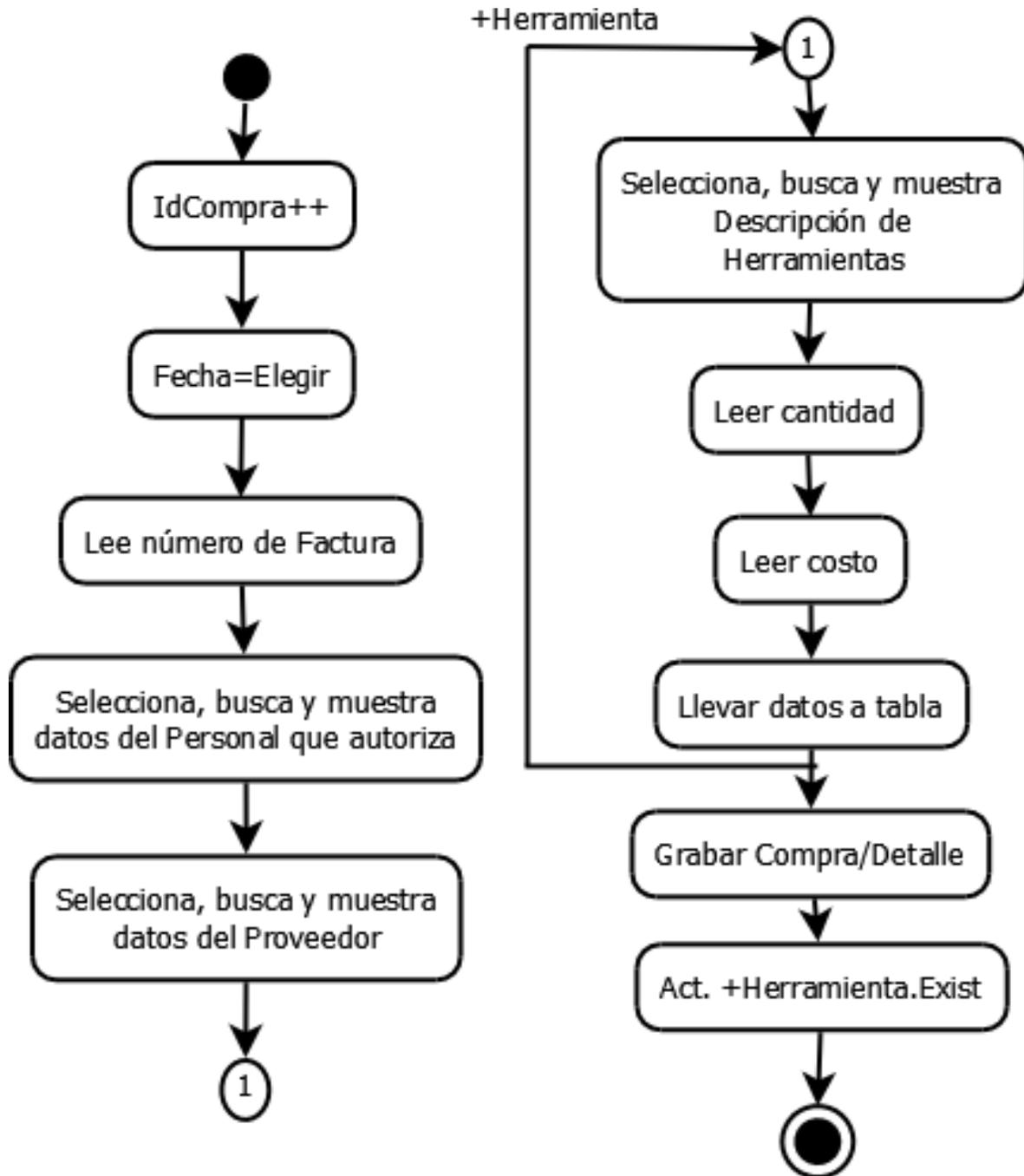


Figura 66. Diagrama de estados para Compras.

## 7.2.7 Diagramas de Reportes

A continuación, se muestran los diagramas utilizados para el movimiento de *Reportes* para las aplicaciones Web y móvil.

### 7.2.7.1 Diagrama de caso de uso expandido

En la Figura 67 se muestra el diagrama de caso de uso expandido para *Reportes*.

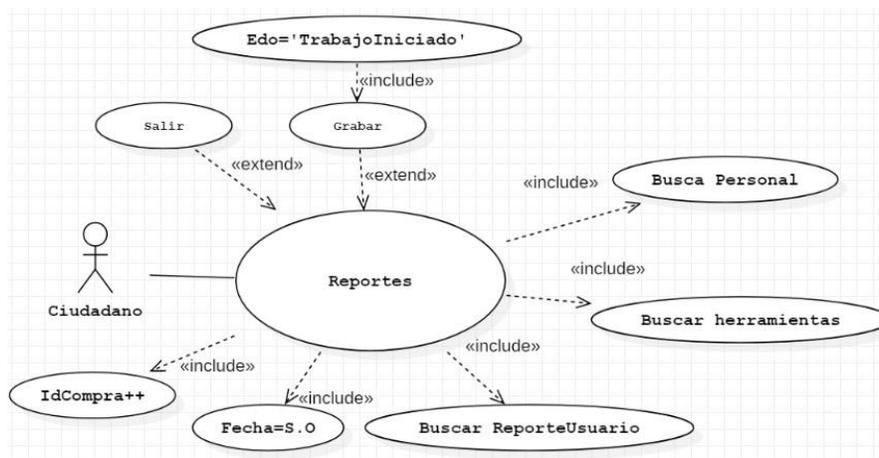


Figura 67. Diagrama de caso de uso expandido para Reportes

### 7.2.7.2 Diagrama de caso de flujo de datos

En la Figura 68 se muestra el diagrama de flujo de datos del movimiento *Reportes*.

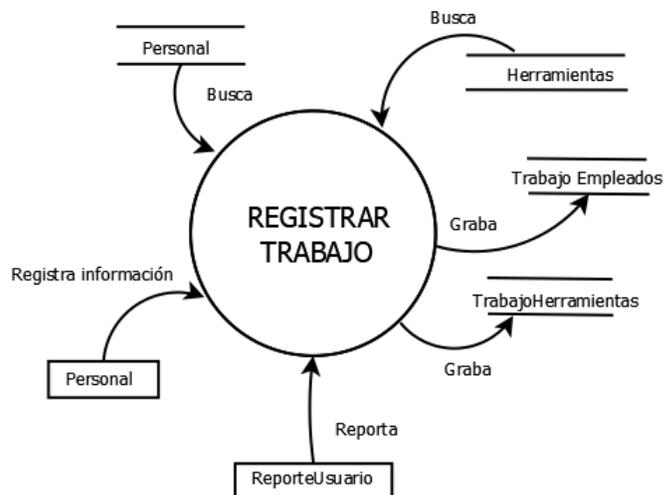


Figura 68. Diagrama de flujo de datos del movimiento Reportes.

### 7.2.7.3 Diagrama de colaboración

En la Figura 69 se muestra el diagrama de colaboración para el caso de uso *Registrar Reporte*.

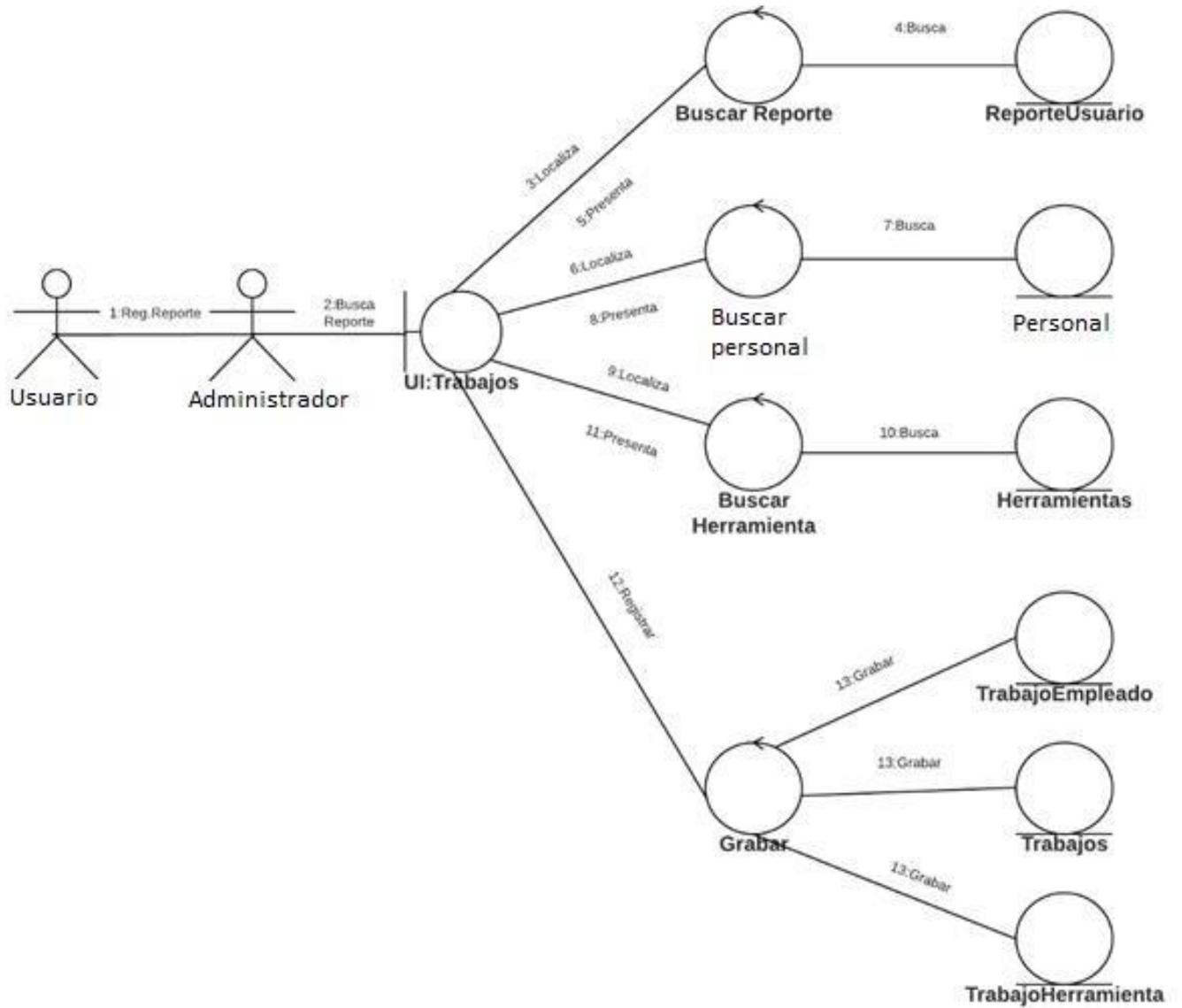


Figura 69. Diagrama de colaboración para Registrar Reportes.

#### 7.2.7.4 Diagrama de estados

En la Figura 70 se muestra el diagrama de estados para el caso de uso *Registrar Reportes*.

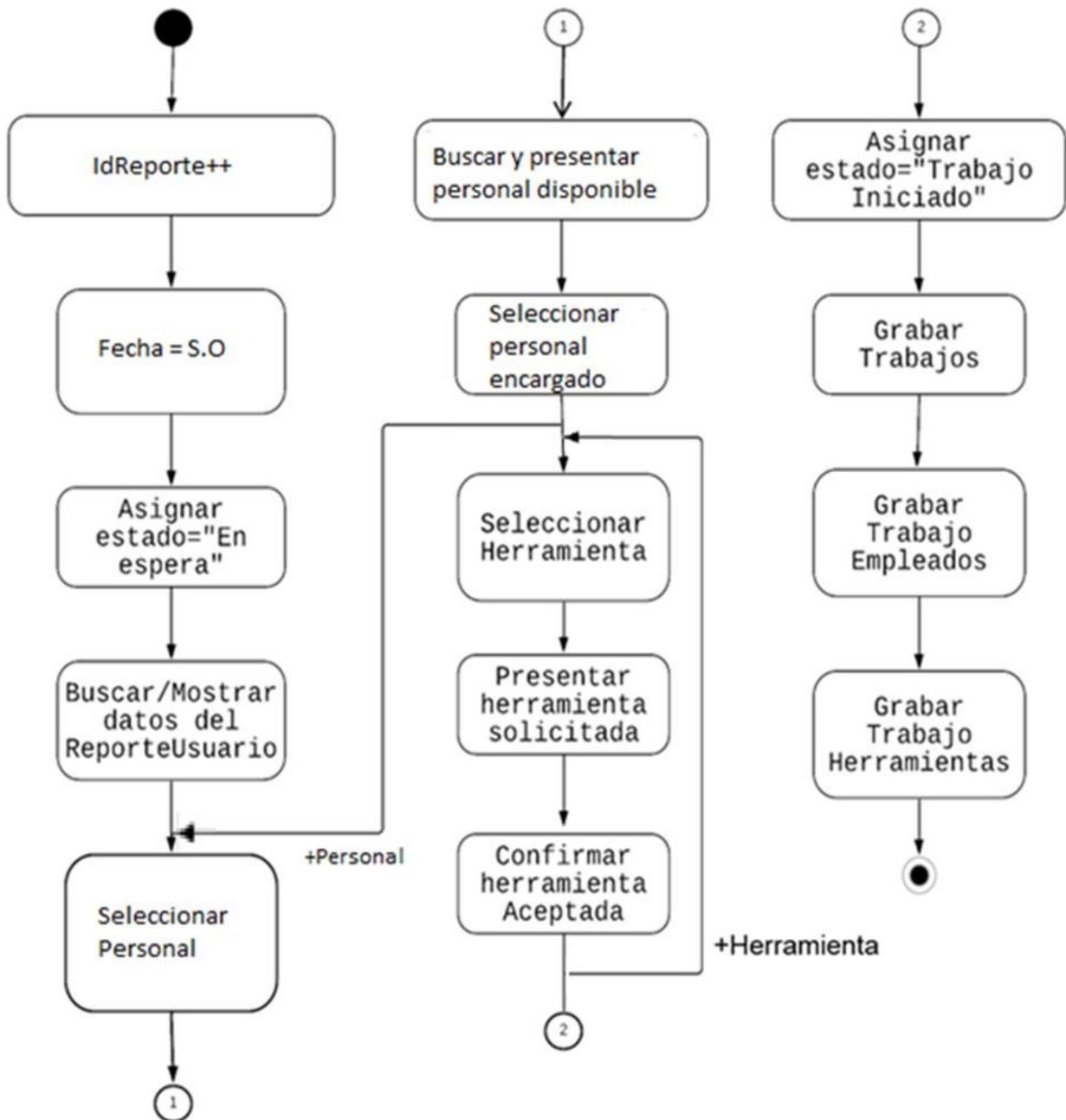


Figura 70. Diagrama de estados para Registrar Reportes.

## 7.2.8 Diagramas de Bajas de herramientas

A continuación, se muestran los diagramas utilizados para el movimiento *Baja de Herramientas*, que se encuentra en la Web.

### 7.2.8.1 Diagrama de caso de uso expandido

En la Figura 71 se muestra el diagrama de caso de uso expandido para baja de herramientas.

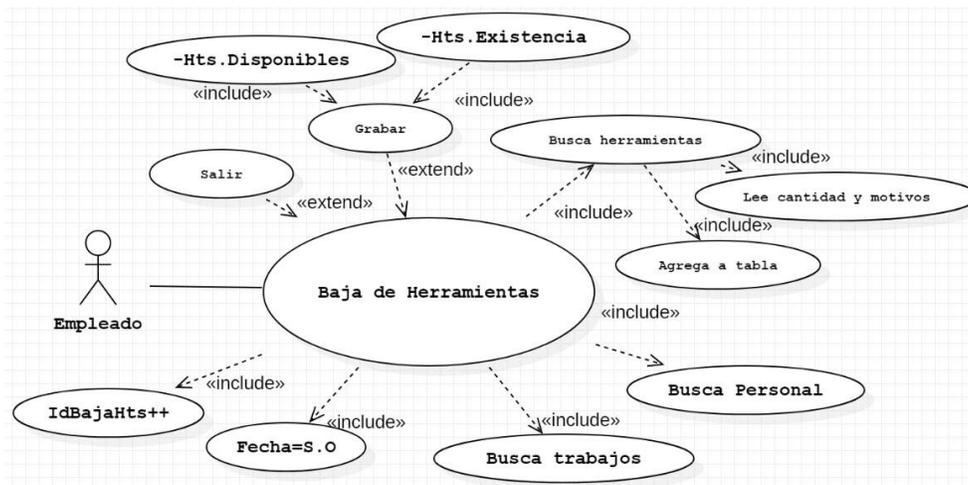


Figura 71. Diagrama de caso de uso expandido para bajas de herramientas.

### 7.2.8.2 Diagrama de flujo de datos

En la Figura 72 se ve el diagrama de flujo de datos para el caso de uso Baja de Herramientas.



Figura 72. Diagrama de flujo de datos para registrar bajas de herramientas.

### 7.2.8.3 Diagrama de colaboración

En la Figura 73 se muestra el diagrama de colaboración para el caso de uso Baja de Herramientas.

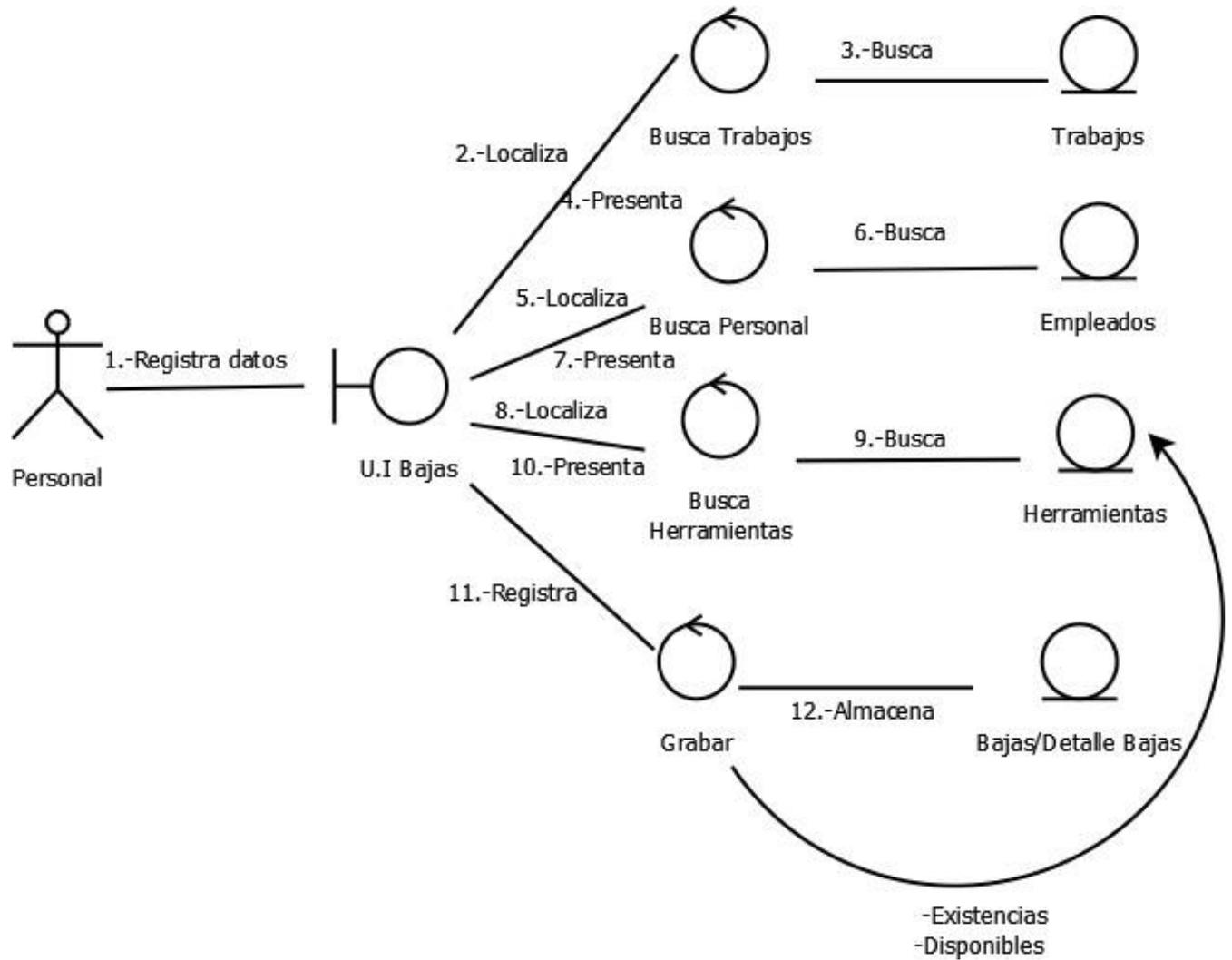


Figura 73. Diagrama de colaboración para bajas de herramientas.

#### 7.2.8.4 Diagrama de estados

En la Figura 74 se muestra el diagrama de estados para el caso de uso Baja de Herramientas.

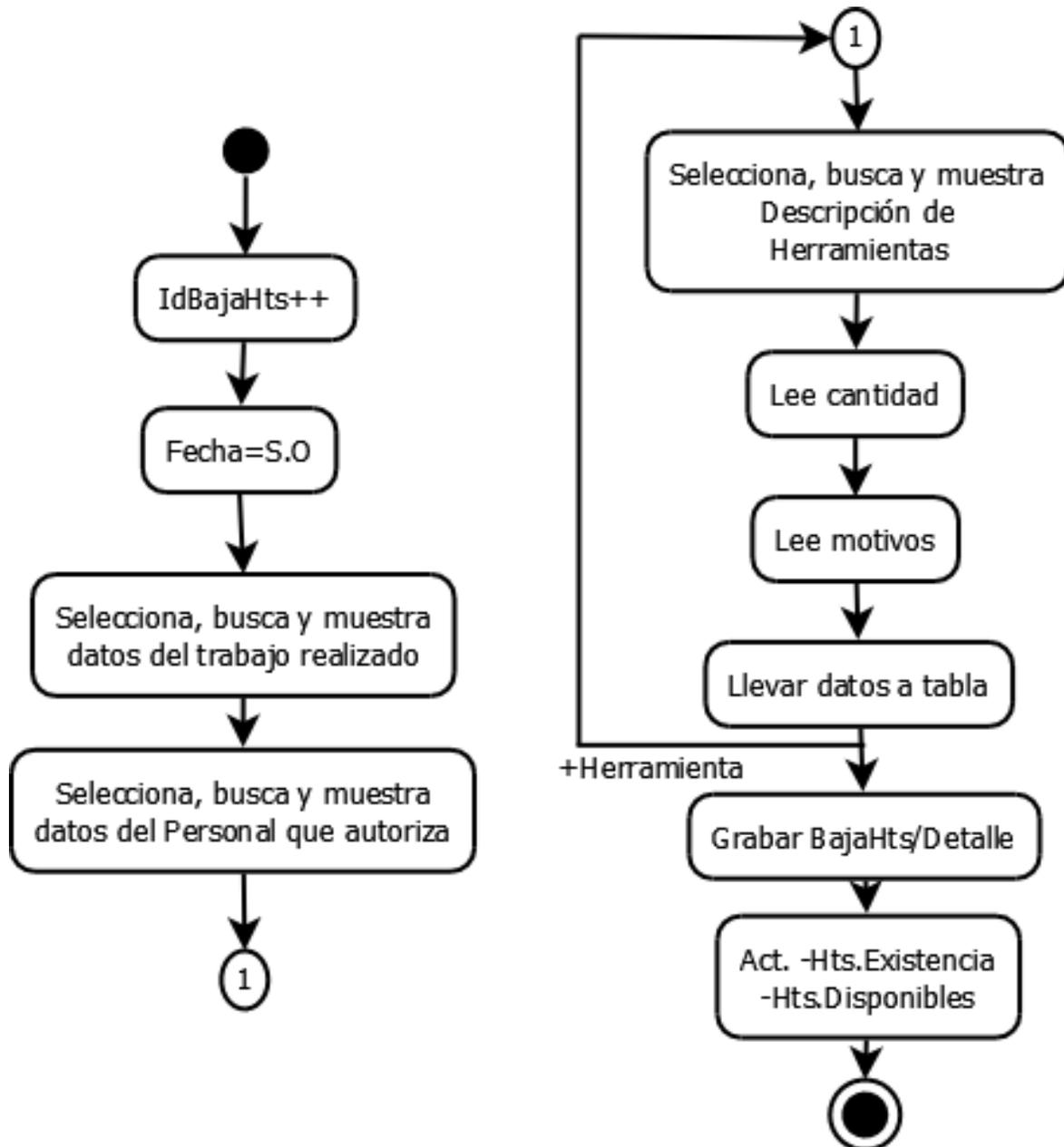


Figura 74. Diagrama de estados para Registrar Bajas.

## 7.2.9 Diagramas de Devoluciones de Herramientas

A continuación, se muestran los diagramas utilizados para el movimiento *Devolución de Herramientas*, el cual se encuentra en la Web.

### 7.2.9.1 Diagrama de caso de uso expandido

En la Figura 75 se ve el diagrama de caso de uso expandido para devolución de herramientas.



Figura 75. Diagrama de caso de uso expandido para Registrar Devoluciones.

### 7.2.9.2 Diagrama de flujo de datos

En la Figura 76 se muestra el diagrama de flujo de datos para *Devolución de Herramientas*.

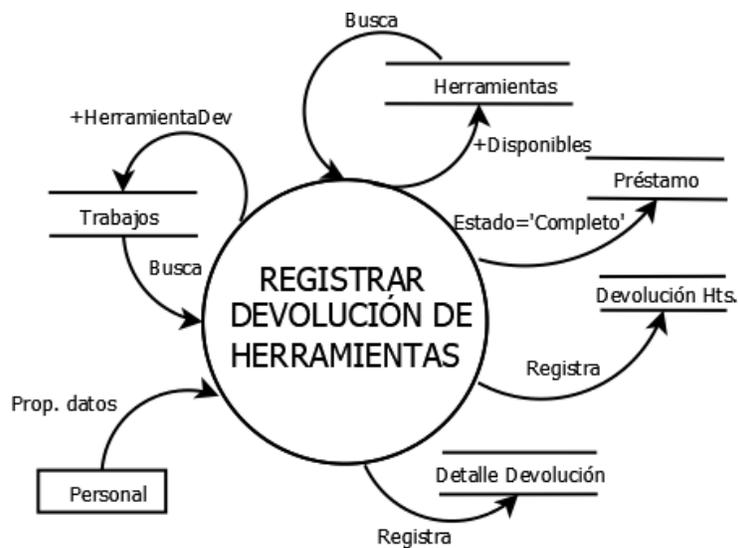


Figura 76. Diagrama de flujo de datos para Devoluciones de Herramientas.

### 7.2.9.3 Diagrama de colaboración

En la Figura 77 se muestra el diagrama de colaboración para el caso de uso *Devolución de Herramientas*.

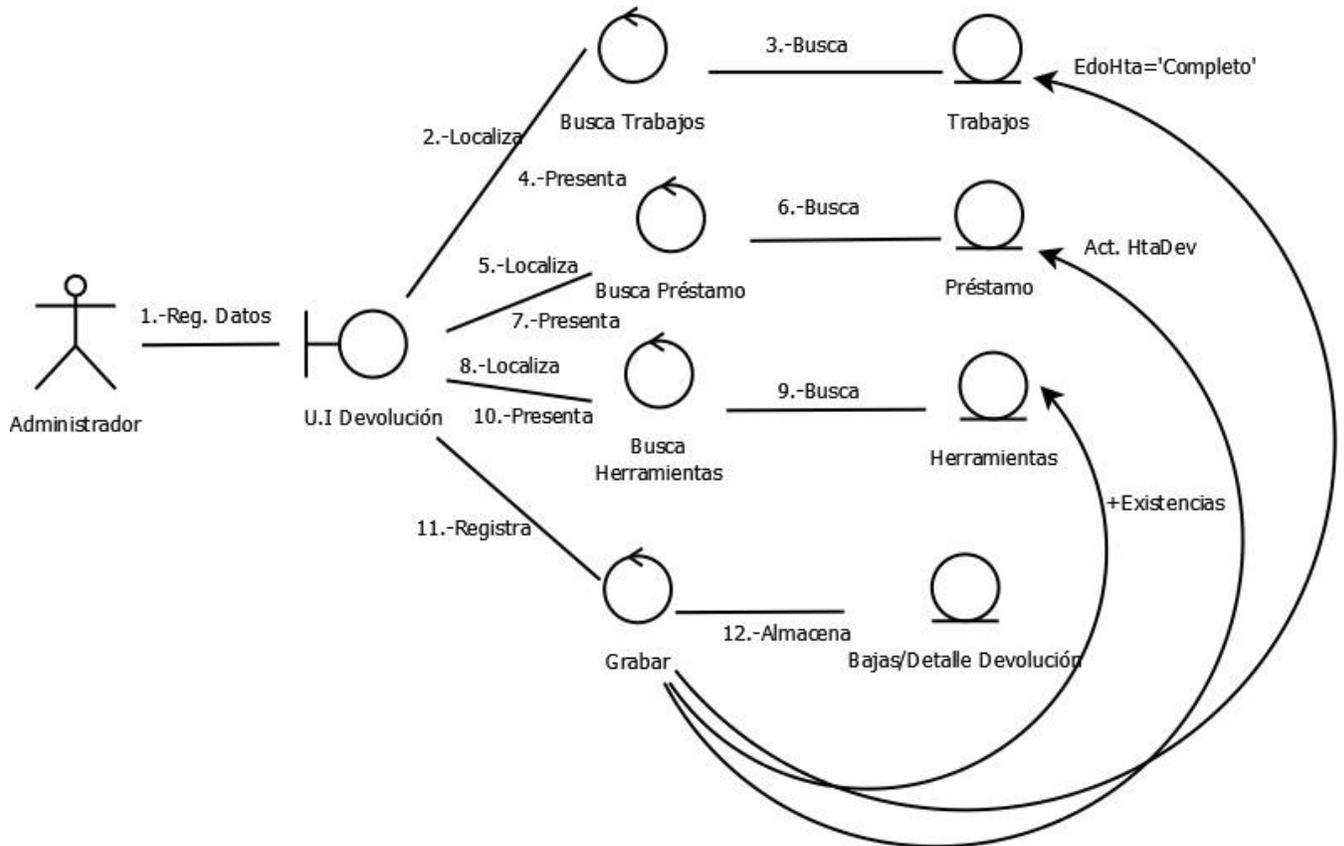


Figura 77. Diagrama de colaboración para Devoluciones de Herramientas.

#### 7.2.9.4 Diagrama de estados

En la Figura 78 se muestra el diagrama de estados para el caso de uso *Devolución de Herramientas*.

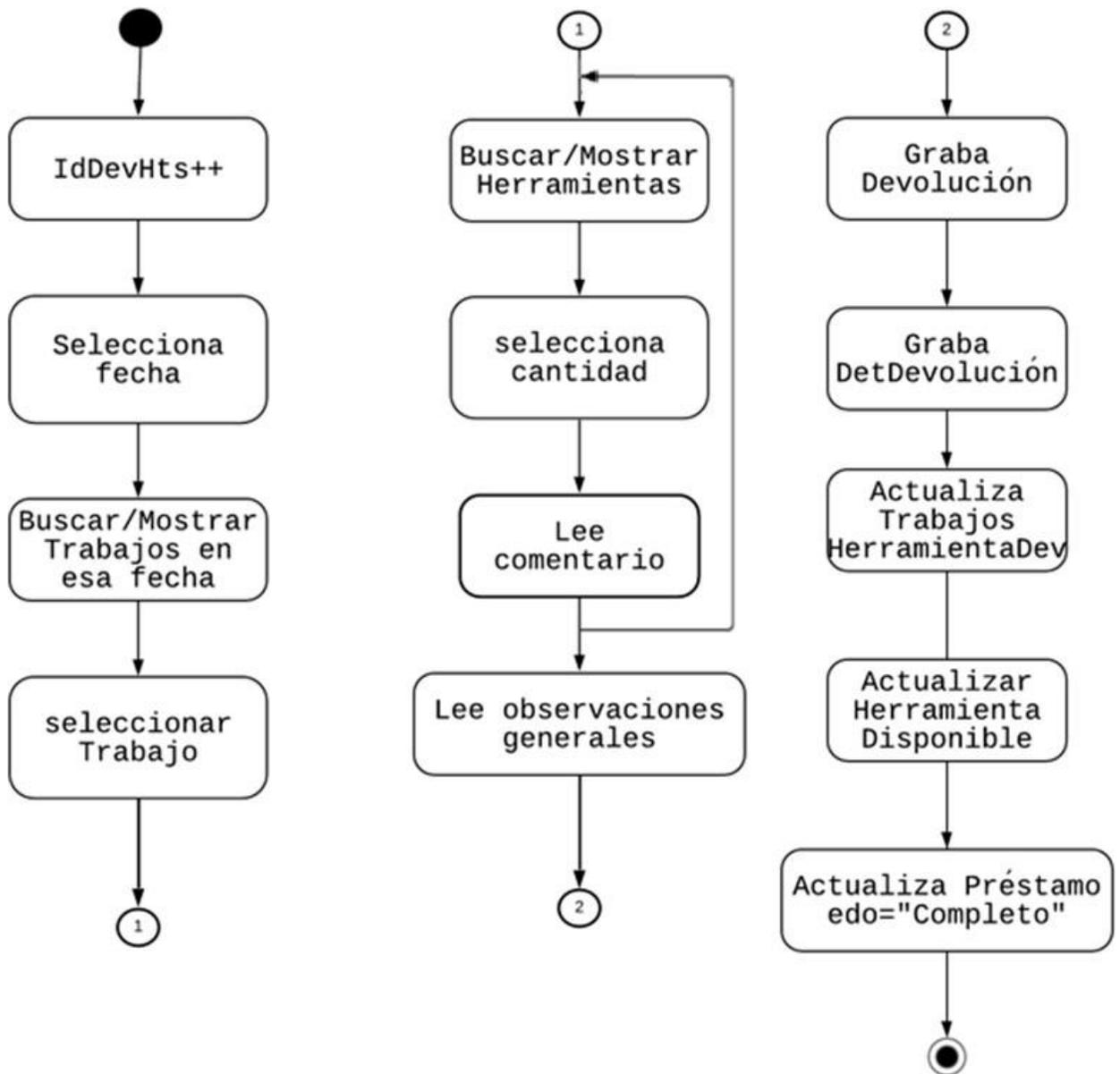


Figura 78. Diagrama de estados para la Devolución de Herramientas.

## 7.3 Pruebas

En la siguiente sección se muestra la parte de pruebas realizadas.

### 7.3.1 Caso de prueba: Movimiento Compras

En la Tabla 1 se muestra la prueba para realizar un *Registro de Compra*.

| <b>Nombre</b>              | <b>Compras</b>  |
|----------------------------|---|
| <i>Descripción</i>         | Registrar una compra de 3 Herramientas  |
| <i>Datos de Entrada</i>    | IdCompra: 22<br>Fecha: 31/08/2020<br>Factura: 01525750<br>Proveedor: Pérez de la Mora<br>Personal que autoriza: Pablo Villegas Gómez<br>Herramienta 1: Pinzas<br>Cantidad: 2<br>Costo: 99<br>Herramienta 2: Serrucho Japonés<br>Cantidad: 3<br>Costo: 200<br>Herramienta 3: Martillo de acero<br>Cantidad: 5<br>Costo: 149  |
| <i>Condiciones</i>         | Serán tres herramientas diferentes, con cantidades y costos diferentes.   |
| <i>Secuencia Principal</i> | <ol style="list-style-type: none"><li>1. Seleccionar la fecha.</li><li>2. Ingresar la factura de la compra.</li><li>3. Seleccionar el proveedor al que se le compraron las herramientas.</li><li>4. Seleccionar el personal que autoriza la operación.</li><li>5. Presionar el botón agregar herramienta.</li><li>6. Seleccionar la herramienta 1.</li><li>7. Ingresar la cantidad.</li><li>8. Ingresar el costo.</li><li>9. Presionar el botón guardar.</li><li>10. Presionar el botón agregar herramienta.</li><li>11. Seleccionar la herramienta 2.</li><li>12. Ingresar la cantidad.</li><li>13. Ingresar el costo.</li><li>14. Seleccionar la herramienta 3</li><li>15. Ingresar la cantidad</li></ol> |

Salida

16. Ingresa el costo
17. Presionar el botón guardar.
18. Presionar el botón grabar datos.

Subtotal: \$1543.00  
Total: \$1789.88  
Graba en compras.  
Actualiza cantidad en herramientas.  
Cantidad final de la herramienta Pinzas:  $14+2 = 16$   
Cantidad final de la herramienta Serrucho Japonés:  
 $8+3 = 11$   
Cantidad final de la herramienta Martillo de acero:  
 $16+5= 21$

Tabla 1. Prueba del movimiento Compras.

En la Figura 79 se muestra la interfaz del movimiento Compras con los datos de la prueba de la Tabla 1.

The screenshot shows a web interface titled "COMPRAS". It contains several form fields and a table. The form fields include: "Fecha" (31/08/2020), "IdCompra" (22), "Factura" (01525750), "Proveedor" (Pérez de la Mora), "Personal que autoriza" (Yael Alejandro Santana), "IdProveedor" (4), "IdPersonal que autoriza" (1), and "Agregar Herramienta" button. The table below has columns: IdHerramienta, Descripción, Cantidad, Costo, Importe, Editar, and Borrar. The table contains three rows of data. At the bottom right, there are fields for "Subtotal" (1543) and "Total" (1789.88).

| IdHerramienta | Descripción       | Cantidad | Costo | Importe | Editar | Borrar   |
|---------------|-------------------|----------|-------|---------|--------|----------|
| 2             | Pinzas            | 2        | 99    | 198     | Ver    | Eliminar |
| 20            | Serrucho Japonés  | 3        | 200   | 600     | Ver    | Eliminar |
| 7             | Martillo de Acero | 5        | 149   | 745     | Ver    | Eliminar |

Figura 79. Interfaz de la prueba del movimiento Compras.

### 7.3.2 Caso de prueba: Baja de Herramientas

En la Tabla 2 se muestra la prueba para registrar la *Baja de Herramientas*.

| <b>Nombre</b>              | <i>Baja de herramientas</i>  |
|----------------------------|--|
| <i>Descripción</i>         | Registrar la baja de 3 herramientas  |
| <i>Datos de Entrada</i>    | IdBaja: 16<br>Fecha: 30/08/2020<br>Calle del trabajo: Prof. Manuel Chávez Madrueño<br>IdTrabajo: 2<br>Personal que autoriza: José Ricardo Velasco Robledo<br>Herramienta 1: Pala punta de huevo<br>Cantidad: 1<br>Motivo: Se rompió<br>Herramienta 2: Nivel<br>Cantidad: 1<br>Motivo: Se desajustó y ya no es preciso.<br>Herramienta 3: Bailejo<br>Cantidad: 1<br>Motivo: Desgaste por trabajo<br>Comentarios generales: Ninguna observación  |
| <i>Condiciones</i>         | Se darán de baja 3 herramientas correspondientes al trabajo con IdTrabajo 2. Estas serán de la misma cantidad, pero con motivo diferente en cada una   |
| <i>Secuencia Principal</i> | <ol style="list-style-type: none"> <li>1. Seleccionar la fecha.</li> <li>2. Selecciona el trabajo al que corresponde la herramienta que será dada de baja.</li> <li>3. Seleccionar el personal que autoriza la operación.</li> <li>4. Presionar el botón agregar herramienta.</li> <li>5. Seleccionar la herramienta 1.</li> <li>6. Ingresa la cantidad.</li> <li>7. Ingresa el motivo.</li> <li>8. Presionar el botón guardar.</li> <li>9. Presionar el botón agregar herramienta.</li> <li>10. Seleccionar la herramienta 2.</li> <li>11. Ingresar la cantidad.</li> <li>12. Ingresar el motivo.</li> <li>13. Selecciona la herramienta 3.</li> <li>14. Ingresa la cantidad.</li> <li>15. Ingresa el motivo.</li> <li>16. Ingresa comentarios generales.</li> <li>17. Presionar el botón guardar.</li> <li>18. Presionar el botón grabar datos.</li> </ol> |

Salida

Graba en Bajas.  
Graba en detalle bajas.  
Actualiza disponibilidad de herramientas.  
Actualiza existencia de herramientas.  
Existencia final de la herramienta Pala punta de  
huevo:  $16-1 = 15$   
Existencia final de la herramienta Nivel:  $23-1 = 22$   
Existencia final de la herramienta Bailejo:  $9-1 = 8$

Tabla 2. Prueba del movimiento baja de herramientas.

En la Figura 80 se muestra la interfaz del movimiento registra *Baja de Herramientas* con los datos de la prueba de la Tabla 2.

**BAJA DE HERRAMIENTAS**

Fecha:  

IdBaja:

Selecciona la calle del trabajo:

IdTrabajo:

Personal que autoriza:

IdPersonal que autoriza:

| IdHerramienta | Descripción         | Cantidad | Motivo                          | Editar                             | Borrar                                  |
|---------------|---------------------|----------|---------------------------------|------------------------------------|---|
| 15            | Pala punta de huevo | 1        | Se rompió                       | <input type="button" value="Ver"/> | <input type="button" value="Eliminar"/> |
| 11            | Nivel               | 1        | Se desajustó y ya no es preciso | <input type="button" value="Ver"/> | <input type="button" value="Eliminar"/> |
| 13            | Bailejo             | 1        | Desgaste por trabajo            | <input type="button" value="Ver"/> | <input type="button" value="Eliminar"/> |

Comentarios generales:

Figura 80. Interfaz de la prueba baja de herramientas.

## 7.4 Pruebas en la aplicación móvil

En la esta sección se presenta la prueba del movimiento *Reportes*, el cual está en ambas plataformas (Web y móvil).

### 7.4.1 Caso de prueba: Registrar Reportes

En la Tabla 2 se muestra la prueba para registrar un *Reporte* desde la cuenta de un usuario en la aplicación móvil.

| <b>Nombre</b>              | <b>Reporte</b>  |
|----------------------------|---|
| <i>Descripción</i>         | Realizar un reporte desde la aplicación móvil.  |
| <i>Datos de Entrada</i>    | IdReporte: 54<br>Fecha: 01/09/2020<br>Tipo de reporte: Bacheo<br>Ubicación: Av. 1ro de Mayo #234<br>Imagen: Foto del carrete.<br>Observaciones: Bache enorme por la avenida primero de mayo frente a una casa.  |
| <i>Condiciones</i>         | Se realiza un reporte de bacheo, el cual incluirá una imagen de prueba que será seleccionada de la galería y estará en la ubicación sobre la avenida mencionada. <ol style="list-style-type: none"> <li>1. Selecciona el tipo de reporte</li> <li>2. Adjunta imagen del carrete -o cámara, en su caso-.</li> <li>3. Selecciona ubicación.</li> <li>4. Ingresa observaciones del reporte.</li> <li>5. Presionar el botón siguiente.</li> <li>6. Se muestra vista general del reporte en otra actividad.</li> </ol> |
| <i>Secuencia Principal</i> | <ol style="list-style-type: none"> <li>7. Presionar el botón confirmar.</li> </ol>  |
| <i>Salida</i>              | Graba en Reporte Usuario.<br>Graba en detalle Reporte Usuario.<br>Guarda imagen en servidor.<br>Guarda coordenadas de la dirección.<br>Actualiza existencia de herramientas.  |

Tabla 3. Prueba del movimiento Reporte.

En la Figura 81 se muestra la interfaz del movimiento *Registrar Reporte* con los datos de la prueba de la Tabla 3



Figura 81. Interfaz de la prueba Registrar Reporte.

## 7.5 Implantación

El método de implantación que se realiza es en paralelo, es decir, que se usan ambos sistemas mientras que el nuevo sistema demuestre ser confiable a su funcionalidad. A continuación, se presenta un cronograma de las actividades a realizar para la implantación en la Tabla 4.

| OCTUBRE 2020  |       |                              |                               |                                |                                |                               |
|---|-------|------------------------------|-------------------------------|--------------------------------|--------------------------------|-------------------------------|
| Actividad   | Fecha | 1                            | 2                             | 3                              | 4                              | 5                             |
|   |       | 1/10/2020<br>al<br>3/10/2020 | 5/09/2020<br>al<br>10/09/2020 | 12/09/2020<br>al<br>17/09/2020 | 19/09/2020<br>al<br>24/09/2020 | 28/09/2020<br>-<br>30/09/2020 |
| Recabar información sobre los empleados, Herramientas y Proveedores |       |                              |                               |                                |                                |                               |
| Capturar Catálogos  |       |                              |                               |                                |                                |                               |
| Probar Módulos  |       |                              |                               |                                |                                |                               |
| Capacitar empleados   |       |                              |                               |                                |                                |                               |
| Revisar Reportes con inventario físico y pendientes                 |       |                              |                               |                                |                                |                               |
| Realizar ajustes requeridos a la parte de diseño o corregir errores |       |                              |                               |                                |                                |                               |
| Liberar Sistema   |       |                              |                               |                                |                                |                               |

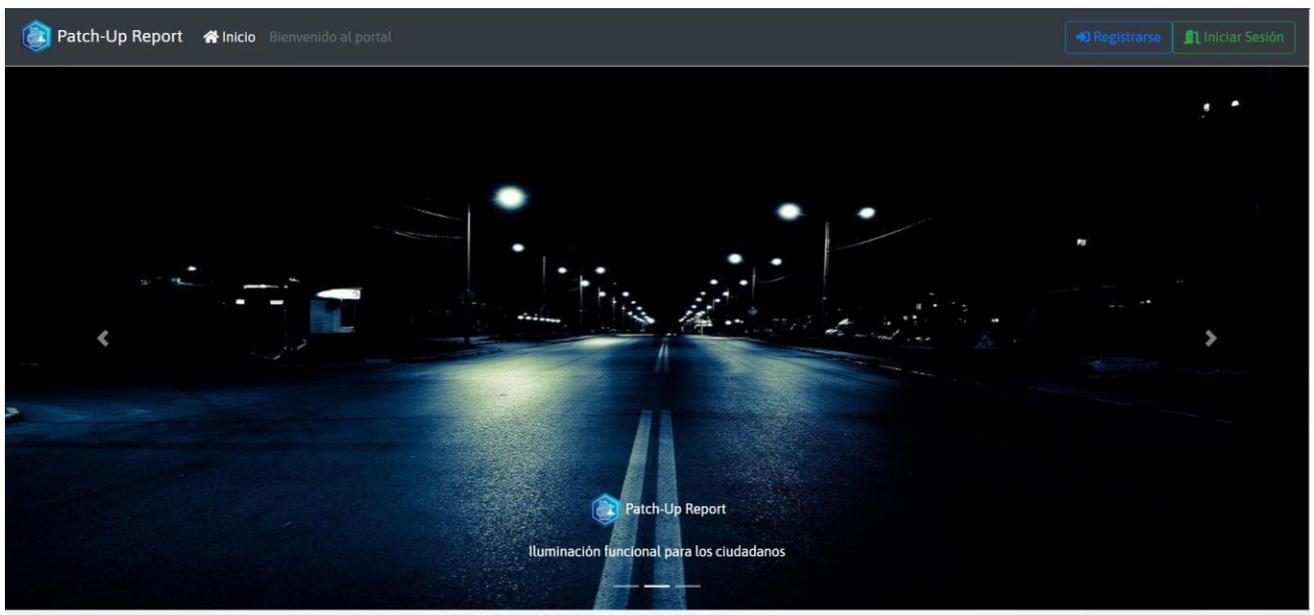
Tabla 4. Cronograma de implantación de la plataforma

## Capítulo 8. Resultados

En esta sección se muestran los resultados del proyecto en cada uno de los dos escenarios: aplicación Web y aplicación móvil.

### 8.1 Sistema Web

En la Figura 82 se muestra la página principal del sistema Web disponible para los *Ciudadanos*.



*Figura 82. Página principal del sistema Web.*

## 8.1.1 Administrador

### 8.1.1.1 Menú

En la Figura 83 se muestra el menú principal para el *Administrador*. Dentro de la opción catálogos se muestran los submenús con las opciones disponibles.

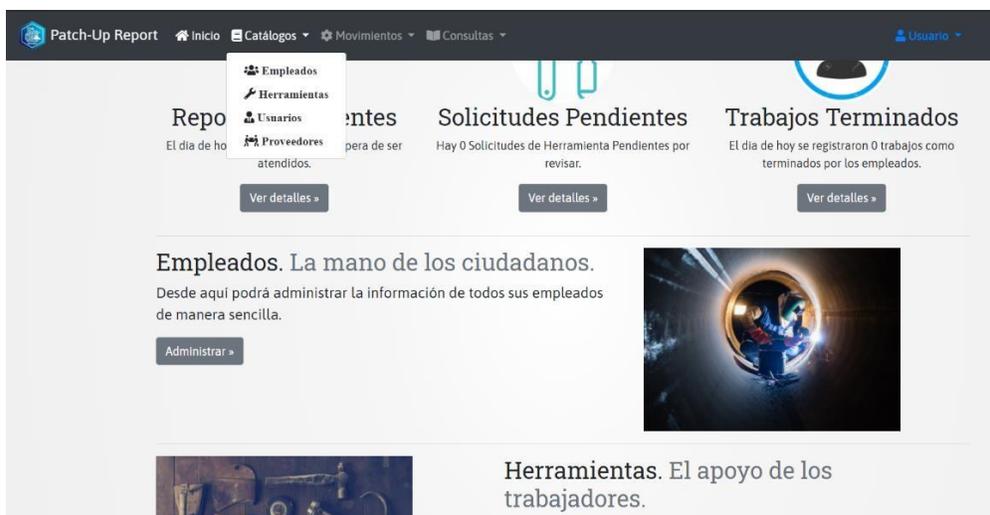


Figura 83. Menú Administrador - Catálogos.

En la Figura 84 se muestra el menú principal para el *Administrador*. Dentro de la opción movimientos se muestran los submenús con las opciones correspondientes.



Figura 84. Menú Administrador - Movimientos.

En la Figura 85 se muestra el menú principal para el *Administrador*. Dentro de la opción consultas se muestran los submenús con las opciones disponibles.



Figura 85. Menú Administrador - Consultas.

### 8.1.1.2 Catálogos

En la Figura 86 se muestra el catálogo de *Usuarios*(ciudadanos).

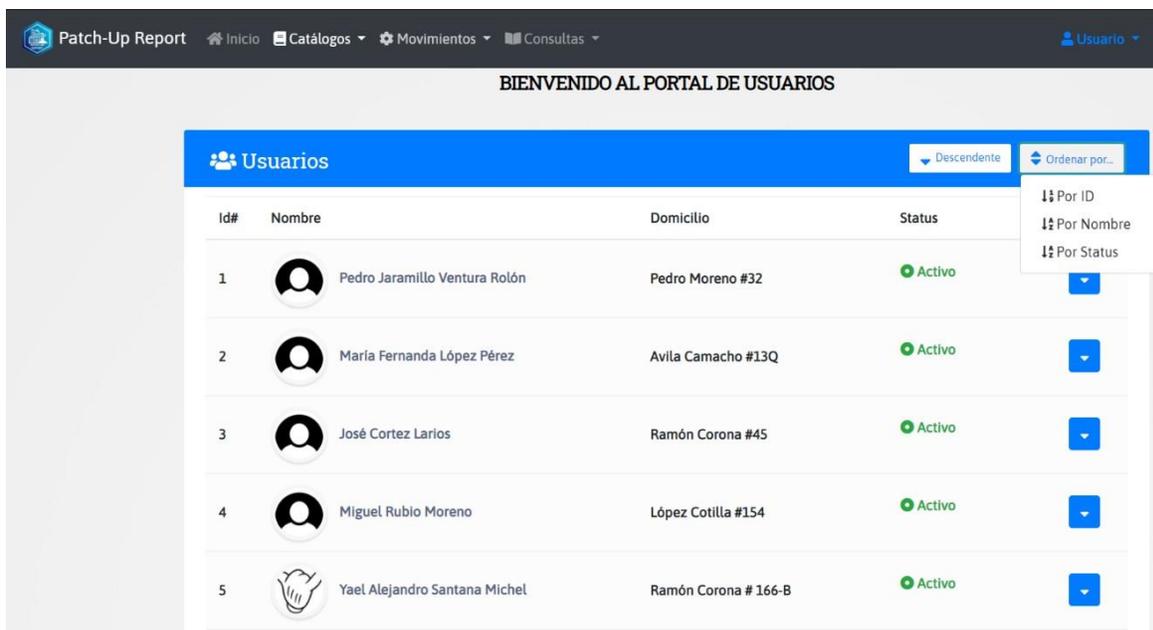


Figura 86. Catálogo de Usuarios.

En la Figura 87 se muestra la opción cambiar estatus dentro del catálogo *Usuarios*.

Nombre  
Pedro Jaramillo Ventura Rolón

Status  
Activo

Activo  
Amonestado  
Suspendido

Guardar Status Cancelar

Figura 87. Editar estatus de Usuarios.

En la Figura 88 se muestra el catálogo *Proveedores*.

BIENVENIDO AL PORTAL DE PROVEEDORES

Proveedores Agregar Proveedor

| IdProveedor | Nombre           | Domicilio        | Colonia | CP    | Teléfono   | Celular    | Ciudad        | Entidad | RFC           | Email              |
|-------------|------------------|------------------|---------|-------|------------|------------|---------------|---------|---------------|--------------------|
| 1           | García de Alba   | Coloón 22        | Centro  | 49000 | 3414124585 | 3411235547 | Ciudad Guzmán | Jalisco | QWERTYUJN11   | gardalba@gmail.com |
| 2           | Mexichem Prueba  | Benito Juárez 62 | Reforma | 49085 | 3411228595 | 34112536   | Ciudad Guzmán | Jalisco | VKSSJAJA      | mexichem@yoprr     |
| 3           | Denise Dresser   | Colón 222        | Centro  | 45820 | 3412589012 | 3579521473 | Ciudad Guzmán | Jalisco | DED91911      | denisedresser@yc   |
| 4           | Pérez de la Mora | Ramón Corona 50  | Centro  | 49000 | 3414135898 | 3411152890 | Ciudad Guzmán | Jalisco | PEDE9806216F4 | pdelamora@gma      |

Figura 88. Catálogo de Proveedores.

En la Figura 89 se muestra la opción para agregar un nuevo *Proveedor*.

REGISTRAR PROVEEDOR

IdProveedor: 5      Nombre:      Domicilio:

Colonia:      C.P.:      Telefono:

Celular:      Ciudad:      Entidad:

R.F.C.:      Email:      Saldo:

Consulta General      Grabar      Cancelar

Figura 89. Registrar Proveedor.

### 8.1.1.3 Movimientos

En la Figura 90 se muestra el movimiento de *Compra de Herramientas*.

COMPRAS

Fecha: dd/mm/aaaa      IdCompra: 18

Factura:

Proveedor: Mexichem Prueba      IdProveedor: 2      Datos: Mostrar/Ocultar

Personal que autoriza: Manolo Robles Ramirez      IdPersonal que autoriza: 20      Datos: Mostrar/Ocultar

Agregar Herramienta

| IdHerramienta | Descripción        | Cantidad | Costo | Importe | Editar | Borrar   |
|---------------|--------------------|----------|-------|---------|--------|----------|
| 4             | Tijeras para podar | 1        | 299   | 299     | Ver    | Eliminar |

Subtotal: 299

Total: 346.84000000000003

Figura 90. Movimiento Compras.

En la Figura 91 se muestra el movimiento *Devolución de Herramientas*.

### DEVOLUCIÓN DE HERRAMIENTAS

Fecha:

Selecciona la calle del trabajo:

Personal que autoriza:

IdDevolución:

IdTrabajo:  Datos: Most/Ocultar

IdPersonal que autoriza:  Datos: Most/Ocultar

Agregar Herramienta

| IdHerramienta | Descripción | Cantidad | Comentarios           | Editar | Borrar   |
|---------------|-------------|----------|-----------------------|--------|----------|
| 12            | Llana       | 1        | En buenas condiciones | Ver    | Eliminar |

Comentario generales:

Toda la devolución fue en tiempo y forma.

Cancelar
Grabar Datos

Figura 91. Movimiento *Devolución de Herramientas*.

En la Figura 92 se puede ver el movimiento de *Reportes*. Aquí es donde el administrador recibe los reportes de usuarios provenientes del sistema Web o la aplicación móvil y los asigna a un trabajador.

Inicio | Catálogos | Movimientos | Consultas
Usuario

# de Seguimiento: 14



Reportado 1 veces

Observaciones

Hay un bache que abarca aproximadamente el 40% de la calle

Usuario que reportó

5  Yael Alejandro Santana Michel

Status: Activo

Sobre la calle Prolongacion Chamizal entre y ver en el mapa

Tipo de Reporte:

Bacheo

Info del Reporte:

Fecha del Reporte: 2019-09-10 18:26:34.843186

Estado: En Espera

[Convertir en trabajo](#)

Mostrando los 5 reportes más recurrentes Ver Todos

Figura 92. Movimiento de *Reportes*.

En la Figura 93 se muestra el movimiento de *Baja de Herramientas*.

**BAJA DE HERRAMIENTAS**

Fecha: 18/09/2020

IdBaja: 4

Selecciona la calle del trabajo: Primavera - 2019-04-04 21:43:00.000000

IdTrabajo: 1

Datos: Most/Ocultar

Intersección 1: Calle Marcos Gordóa

Intersección 2: Santos Degollado

Observaciones: Hay un Bache Enorme justo en el centro de

Veces Reportado: 1

Latitud: 19.701575

Longitud: -103.459787

Personal que autoriza: Yael Alejandro Santana

IdPersonal que autoriza: 1

Datos: Most/Ocultar

Agregar Herramienta

| IdHerramienta | Descripción | Cantidad | Motivo | Editar | Borrar   |
|---------------|-------------|----------|--------|--------|----------|
| 5             | Machete     | 1        | Rota.  | Ver    | Eliminar |

Comentarios generales:  
El machete se encontraba roto por aparente desgaste de uso.

Figura 93. Movimiento Bajas de Herramienta.

#### 8.1.1.4 Consultas

En la Figura 94 se muestra la consulta de *Compra de Herramientas* filtrada por fechas.

**CONSULTA DE COMPRA DE HERRAMIENTAS**

Selecciona el tipo de consulta:  Por fechas  Por proveedor  Por personal que autoriza

Fecha Inicial: 18/02/2020

Fecha Final: 18/02/2020

Buscar

| IdCompra | Nombre del Proveedor | Domicilio | Empleado que autorizó | Factura         | Fecha      | Total | Seleccionar |
|----------|----------------------|-----------|-----------------------|-----------------|------------|-------|-------------|
| 3        | Denise Dresser       | Colón 222 | Pablo Villegas Gomez  | 8142AQQJSMW1225 | 2020-02-18 | 162.4 | Ver detalle |

**Detalle de la Compra: 3**

| IdHerramienta | Nombre             | Descripción                      | Cantidad | Costo | Importe |
|---------------|--------------------|----------------------------------|----------|-------|---------|
| 1             | Tijeras para podar | Herramienta Para Corte de Pasto  | 2        | 20    | 40      |
| 2             | Pinzas             | Para cortar alambre, clavos, etc | 2        | 50    | 100     |

Total de la compra: \$140 MXN

Figura 94. Consulta de Compras por rango de fechas.

En la Figura 95 se muestra la consulta de *Compra de Herramientas* filtrada por proveedor.

The screenshot shows a web application interface for 'CONSULTA DE COMPRA DE HERRAMIENTAS'. At the top, there is a navigation bar with 'Inicio', 'Catálogos', 'Movimientos', and 'Consultas'. The main content area has a title 'CONSULTA DE COMPRA DE HERRAMIENTAS' and a search filter section. The filter is set to 'Por proveedor' (selected), with 'Pérez de la Mora' selected in the dropdown and a 'Buscar' button. Below the filter is a table with columns: IdCompra, IdEmpleado autorizó, Nombre, Teléfono, Factura, Fecha, Total, and Seleccionar. Two records are shown: one for Yael Alejandro Santana (IdCompra: 7, Total: 6.96) and one for Roberto Alexander Montes Rolón (IdCompra: 9, Total: 249.4). Below the table is a section titled 'Detalle de la Compra: 9' with a sub-table showing items: 'Martillo Electrico Demoedor' (IdHerramienta: 4, Cantidad: 2, Costo: 20, Importe: 40) and 'Pico' (IdHerramienta: 17, Cantidad: 1, Costo: 175, Importe: 175). The total for this purchase is \$215 MXN.

Figura 95. Consulta de Compras por proveedor.

En la Figura 96 se muestra la consulta de *Compra de Herramientas* filtrada por personal que autorizó.

The screenshot shows the same web application interface, but the search filter is set to 'Por personal que autoriza' (selected). The dropdown shows 'Pablo Villegas Gomez' and the 'Buscar' button is present. The table below has columns: IdCompra, IdProveedor, Nombre, Domicilio, Teléfono, Factura, Fecha, Total, and Seleccionar. One record is shown for Denise Dresser (IdCompra: 3, Total: 162.4). Below the table is a section titled 'Detalle de la Compra: 3' with a sub-table showing items: 'Tijeras para podar' (IdHerramienta: 1, Cantidad: 2, Costo: 20, Importe: 40) and 'Pinzas' (IdHerramienta: 2, Cantidad: 2, Costo: 50, Importe: 100). The total for this purchase is \$140 MXN.

Figura 96. Consulta de Compras por personal que autoriza.

En la Figura 97 se muestra la consulta de *Devolución de Herramientas* filtrada por trabajos.

Figura 97. Consulta de Devoluciones por trabajo.

En la Figura 98 se muestra la consulta de *Baja de herramientas* filtrada por fechas.

| IdBaja | IdEmpleado | Nombre                   | Calle trabajo                      | Observaciones usuario                          | Fecha Reporte              | Estado           | Fecha Baja | Comentarios    | Seleccionar                 |
|--------|------------|--------------------------|------------------------------------|--|----------------------------|------------------|------------|----------------|-----------------------------|
| 1      | 6          | Guadalupe Santos Reyes   | Avenida Fed. del Toro              | se está tirando el agua                        | 2019-09-11 16:42:17.498795 | Trabajo Iniciado | 2020-03-19 | Quedan 6-1     | <a href="#">Ver detalle</a> |
| 2      | 2          | Miguel Ángel Vidrio Páez | Calle Prof. Manuel Chávez Madrueño | Hay una Enorme fuga de agua justo en el centro | 2019-04-04 21:43:00.000000 | Trabajo Iniciado | 2020-03-01 | segunda prueba | <a href="#">Ver detalle</a> |

| Detalle de la Baja: 1 |         |  |          |           |
|-----------------------|---------|--|----------|-----------|
| IdHerramienta         | Nombre  | Descripción  | Cantidad | Motivo    |
| 13                    | Bailejo | llana metálica con forma triangular y puntas redondeadas usada para mezclar y colocar mezcla | 1        | se quebró |

Figura 98. Consulta de Bajas por fechas.

En la Figura 99 se muestra la consulta de *Baja de herramientas* filtrada por trabajos.

Selecciona el tipo de consulta:  Por fechas  Por trabajos  Por personal que autoriza

Calle Prof. Manuel Chávez Madrueño - Id:3

Buscar

Intersección 1: Calle Gral. Lázaro Cárdenas del Río

Intersección 2: Calle Alejandro Humboldt

Observaciones: Hay una Enorme fuga de agua justo en el c

Veces Reportado: 4

Latitud: 19.70326

Longitud: -103.458135

| IdBaja | IdEmpleado | Nombre                   | Estado           | Fecha Baja | Comentarios    | Seleccionar                 |
|--------|------------|--------------------------|------------------|------------|----------------|-----------------------------|
| 2      | 2          | Miguel Ángel Vidrio Páez | Trabajo Iniciado | 2020-03-01 | segunda prueba | <a href="#">Ver detalle</a> |

**Detalle de la Baja: 2**

| IdHerramienta | Nombre | Descripción   | Cantidad | Motivo     |
|---------------|--------|---|----------|------------|
| 10            | Barra  | barra de hierro con un lado en punta y el otro plano. Utilizado para excavaciones manuales. | 1        | Se enchucó |

Figura 99. Consulta de Bajas por trabajos.

En la Figura 100 se muestra la consulta de *Baja de herramientas* filtrada por personal que autorizó.

Selecciona el tipo de consulta:  Por fechas  Por trabajos  Por personal que autoriza

Miguel Ángel Vidrio Páez

Buscar

| IdBaja | Calle trabajo                      | Observaciones usuario                          | Fecha Reporte              | Estado           | Fecha Baja | Comentarios    | Seleccionar                 |
|--------|------------------------------------|--|----------------------------|------------------|------------|----------------|-----------------------------|
| 2      | Calle Prof. Manuel Chávez Madrueño | Hay una Enorme fuga de agua justo en el centro | 2019-04-04 21:43:00.000000 | Trabajo Iniciado | 2020-03-01 | segunda prueba | <a href="#">Ver detalle</a> |

**Detalle de la Baja: 2**

| IdHerramienta | Nombre | Descripción   | Cantidad | Motivo     |
|---------------|--------|---|----------|------------|
| 10            | Barra  | barra de hierro con un lado en punta y el otro plano. Utilizado para excavaciones manuales. | 1        | Se enchucó |

Figura 100. Consulta de Bajas por personal que autoriza.

## 8.1.2 Empleado

En la Figura 101 se muestra la vista principal de la aplicación Web para el *Empleado*.

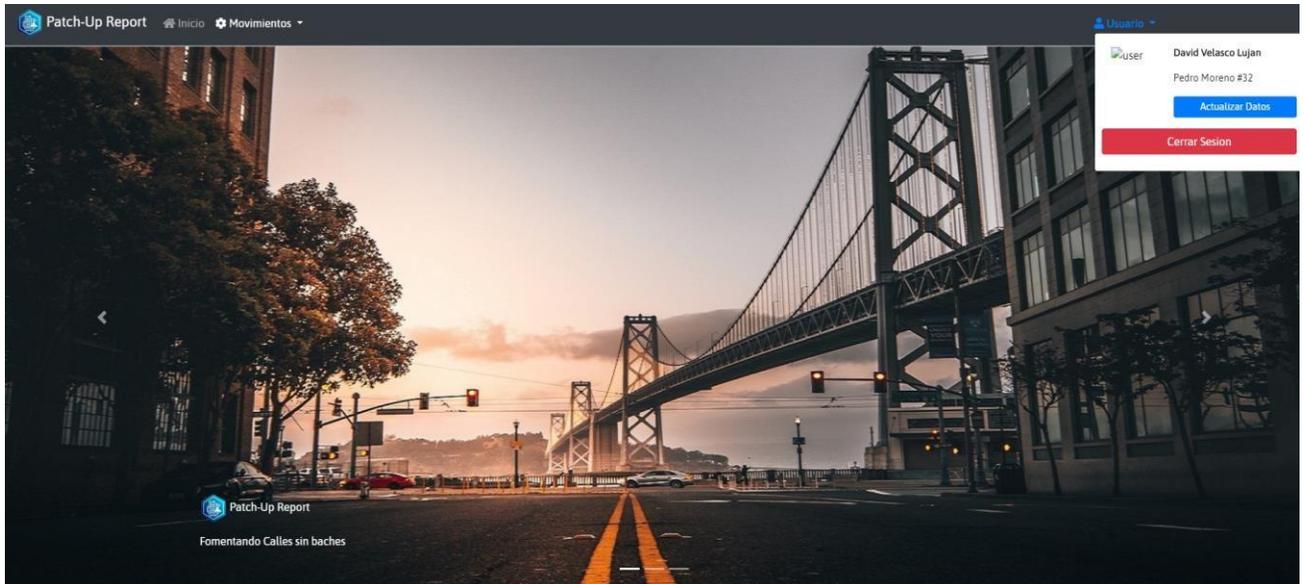


Figura 101. Vista principal del Empleado.

### 8.1.2.1 Menú

En la Figura 102 se muestra el menú para el *Empleado*.

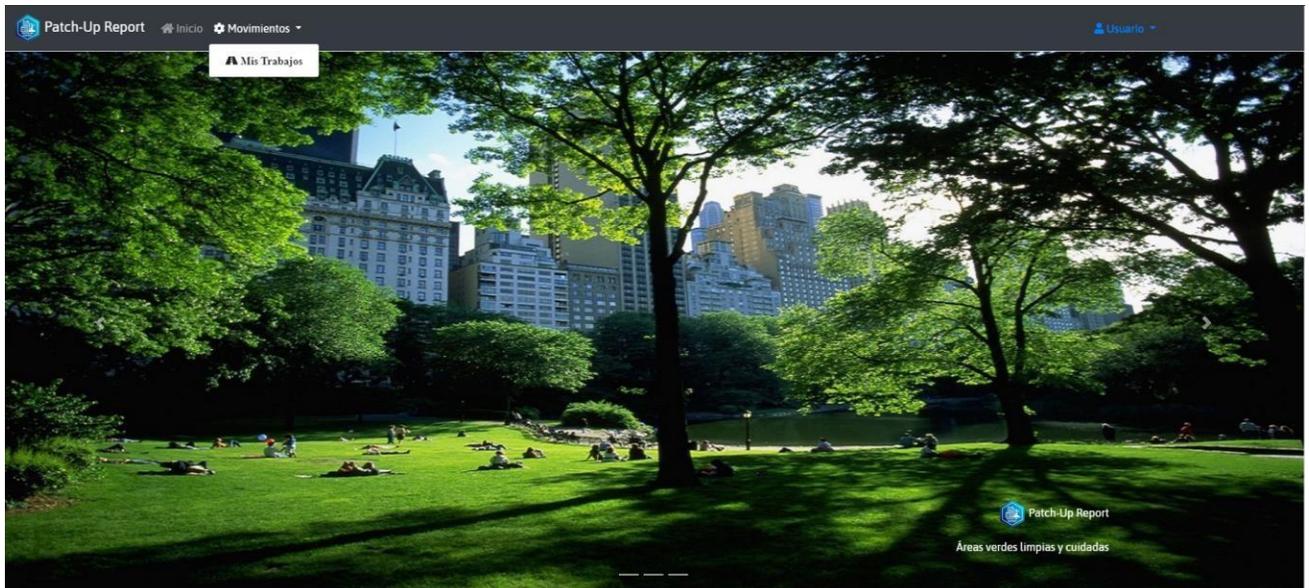


Figura 102. Menú - movimientos Empleado.

### 8.1.2.2 Movimiento

En la Figura 103 se muestra el movimiento de *Reportes* para el perfil *Empleado*. Aquí se presentan los reportes asignados por el administrador al empleado, mismos que a su vez fueron realizados por usuarios a través de la aplicación móvil o el sistema Web.

The screenshot shows a web application interface for 'TRABAJOS REGISTRADOS'. At the top, there is a navigation bar with 'Inicio', 'Catálogos', and 'Movimientos' on the left, and 'Usuario' on the right. Below the navigation bar, the title 'TRABAJOS REGISTRADOS' is centered. The main content area features a blue header with 'Todos los Reportes' and a dropdown menu for 'Ordenar por...'. Below this is a table with the following data:

| Calle                                 | Observaciones           | Estado            |   |
|---------------------------------------|-------------------------|-------------------|---|
| Unnamed Road                          | dónde fin en un osos am | En Espera         | ▼ |
| Calle José María Morelos y Pavón      | Descripción             | Trabajo Terminado | ▼ |
| Calle Gral. Miguel Contreras Medellín | Hay un bache inmensooo  | Trabajo Terminado | ▼ |

At the bottom of the table, it says 'Mostrando 10 de 3 Encontrados' and 'Previous 1 Next'.

Figura 103. Movimiento Reportes empleado.

## 8.2 Aplicación móvil

En esta sección se presenta la aplicación móvil, la cual es de uso exclusivo para el usuario que reporta. Dentro de ella puede registrarse, acceder, realizar un reporte, consultarlos por diferentes métodos y consultar el seguimiento, entre otras opciones.

### 8.2.1 Usuario

Desde este perfil, que es el usuario final, se puede hacer todo lo que se muestra a continuación.

#### 8.2.1.1 Pantalla de inicio de sesión

En la Figura 104 se muestra la pantalla de inicio de sesión de la aplicación.

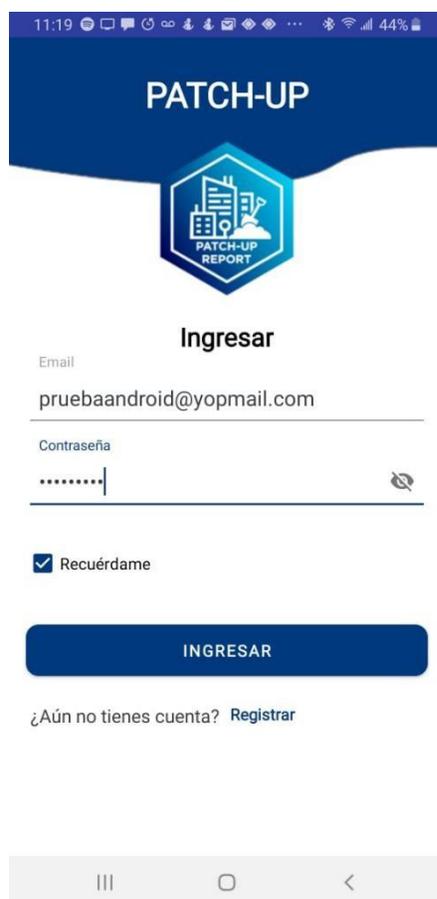


Figura 104. Pantalla de inicio.

### 8.2.1.2 Registrarse

En las Figuras 105 y 106 se muestra la pantalla de registro.



Figura 105. Pantalla de registro.



Figura 106. Pantalla de registro 2.

### 8.2.1.3 Solicitud de permisos.

Cuando se inicia la aplicación por primera vez en un dispositivo, se ejecutan las solicitudes de permisos peligrosos. Ver en las Figuras 107, 108 y 109.

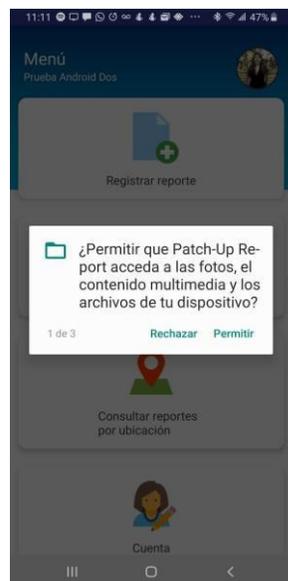


Figura 109. Permiso archivos.

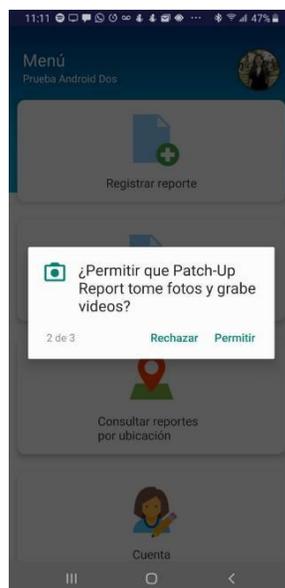


Figura 107. Permiso cámara.

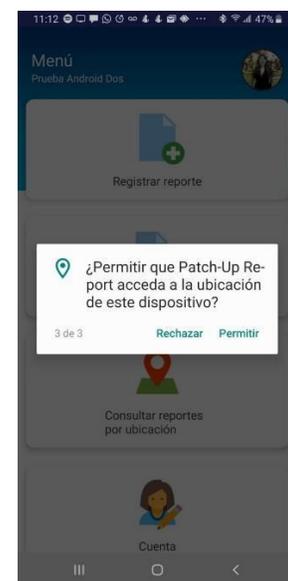


Figura 108. Permiso ubicación.

### 8.2.1.3 Menú principal

En las Figuras 110 y 111 se puede muestra el menú principal con las diferentes opciones.

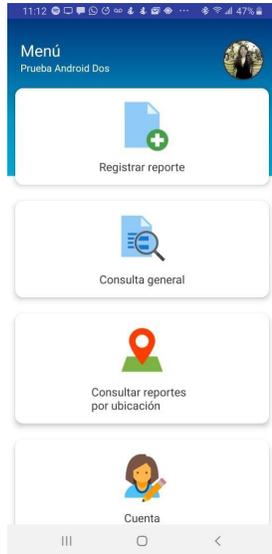


Figura 111. Menú principal 1.

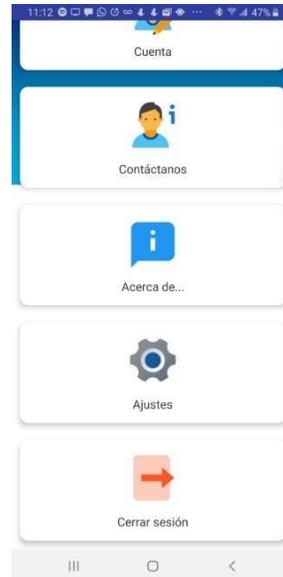


Figura 110. Menú principal 2.

### 8.2.1.4 Movimiento reportar

En las Figuras 112 y 113 se puede ver el movimiento *Reportar*.



Figura 113. Realizar reporte.



Figura 112. Resumen del reporte.

### 8.2.1.5 Consulta general

En la figura 114 se observa la consulta general de reportes realizados por el usuario autenticado.



Figura 114. Consulta general.

### 8.2.1.6 Consulta por ubicación (mapa)

En la Figura 116 se presenta la consulta general en el mapa donde fue hecho el reporte.

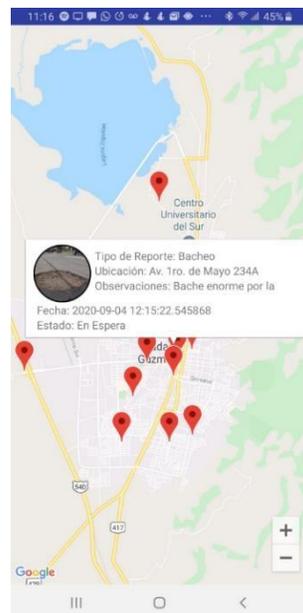


Figura 115. Consulta en mapa.

### 8.2.1.7 Consultar detalles del reporte

Una vez que el reporte fue atendido y solucionado por el personal asignado, se pueden consultar los resultados. Ver en la Figura 116.



Figura 116. Resultados del reporte.

### 8.2.1.8 Actualizar información

Al usuario se le permite actualizar los datos. Ver la pantalla en la Figura 117.



Figura 117. Actualizar información.

### 8.2.1.9 Contáctanos

En las Figuras 118 y 119 se muestran las opciones dentro del menú “Contáctanos”.



Figura 119. Pantalla contacto.



Figura 118. Pantalla contacto 2.

### 8.2.1.10 Equipo

En la Figura 120 se presenta la actividad de la opción “Equipo”

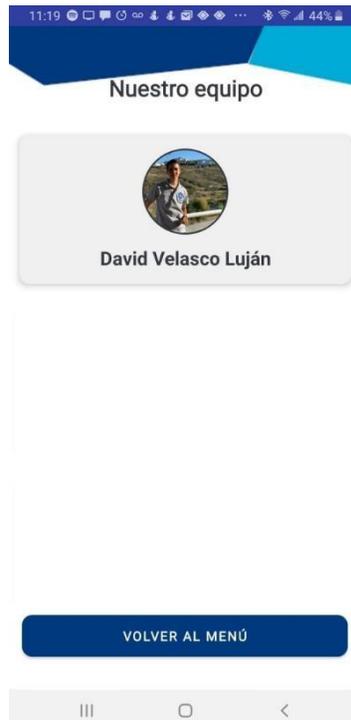


Figura 120. Equipo.

### 8.2.1.11 Ajustes

Como se trata de una aplicación internacionalizada, se puede cambiar el idioma de la aplicación entre español e inglés. En la Figura 121 se presenta la actividad para cambiar el idioma; en la Figura 122 se observa que se cambió al idioma inglés.



Figura 121. Idioma.



Figura 122. Cambio a inglés.

## Capítulo 9. Temas de discusión

Durante el desarrollo del sistema multiplataforma Web y móvil, se presentaron distintos problemas que se muestran a continuación, así como su respectiva solución.

### 9.1 Comunicación entre plataformas Web y móvil

Al principio, se planteó usar el método *HTTP Handler* para comunicar la base de datos que se encuentra en un servidor Web y mostrar y manipular la información desde la aplicación móvil. Se realizó el registro de usuario por este método, pero después, por medio de investigación se encontró la librería Volley, la cual facilita la comunicación entre las plataformas mediante Web services.

Por ejemplo, sin la librería Volley, se tiene que crear un adaptador para ahí almacenar los datos de manera temporal, así como una clase específica para leer y escribir en la base de datos. En la Figura 123 se muestran imágenes del proceso para el registro sin el método.

```
17     private static final String TAG = HttpHandler.class.getSimpleName();
18
19     @
20     public HttpHandler() {
21     }
22
23     public String makeServiceCall(String reqUrl) {
24         String response = null;
25         try {
26             URL url = new URL(reqUrl);
27             HttpURLConnection conn = (HttpURLConnection) url.openConnection();
28             conn.setRequestMethod("GET");
29             // Leyendo la respuesta
30             InputStream in = new BufferedInputStream(conn.getInputStream());
31             response = convertStreamToString(in);
32         } catch (MalformedURLException e) {
33             Log.e(TAG, msg: "MalformedURLException: " + e.getMessage());
34         } catch (ProtocolException e) {
35             Log.e(TAG, msg: "ProtocolException: " + e.getMessage());
36         } catch (IOException e) {
37             Log.e(TAG, msg: "IOException: " + e.getMessage());
38         } catch (Exception e) {
39             Log.e(TAG, msg: "Exception: " + e.getMessage());
40         }
41         return response;
42     }
43
44     @
45     private String convertStreamToString(InputStream is) {
46         BufferedReader reader = new BufferedReader(new InputStreamReader(is));
47         StringBuilder sb = new StringBuilder();
48         String line;
49         try {
50             while ((line = reader.readLine()) != null) {
51                 sb.append(line).append('\n');
52             }
53         } catch (IOException e) {
54             e.printStackTrace();
55         } finally {
56             try {
57                 is.close();
58             } catch (IOException e) {
59                 e.printStackTrace();
60             }
61         }
62         return sb.toString();
63     }
64 }
```

Figura 123. Método HTTP Handler

En la Figura 124 se observa que también se crea un archivo java donde se especifica la ubicación del servicio Web, así como los campos que leerá e insertará en la base de datos una vez que esté preparado.

```
8
9 public interface RegisterAPI {
10     @FormUrlEncoded
11     @POST("/AppAndroid/Registrar.php")
12     public void insertUser(
13         @Field("nombre") String nombre,
14         @Field("domicilio") String domicilio,
15         @Field("telefono") String telefono,
16         @Field("email") String email,
17         @Field("password") String password,
18         Callback<Response> callback);
19
20     public void leerUsuario(
21         @Field("nombre") String nombre,
22         @Field("domicilio") String domicilio,
23         @Field("telefono") String telefono,
24         @Field("email") String email,
25         @Field("password") String password,
26         Callback<Response> callback
27     );
28
29 }
30
```

Figura 124. Archivo API.

En la Figura 125 se muestra la diferencia del proceso, en esa parte se muestra el proceso de inicio de sesión de la aplicación, pero implementando la librería Volley, ahí se puede ver la diferencia, ya que únicamente se hace uso de dos métodos; *onResponse* y *onError*.

```

161 @Override
162 public void onResponse(JSONObject response) {
163     progreso.hide();
164     btIngresar.setEnabled(true);
165     tvRegistrar.setEnabled(true);
166     if (response.length() < 1) {
167         Snackbar snackbar = Snackbar.make(btIngresar, "Correo o contraseña incorrecta." + " " + etEmail.getText().toString(), Snackbar.LENGTH_SHORT);
168         snackbar.show();
169         //Toast.makeText(this, "Correo o contraseña incorrecta: " + etEmail.getText().toString(), Toast.LENGTH_SHORT).show();
170     } else {
171         Snackbar snackbar = Snackbar.make(btIngresar, "Consulta exitosa al servidor." + " " + etEmail.getText().toString(), Snackbar.LENGTH_SHORT);
172         snackbar.show();
173         //Toast.makeText(this, "Inicio de sesión correcto: " + etEmail.getText().toString(), Toast.LENGTH_SHORT).show();
174         Usuario usuario = new Usuario();
175
176         JSONArray jsonArray = response.optJSONArray( name: "datos");
177         JSONObject jsonObject = null;
178         try {
179             jsonObject = jsonArray.getJSONObject( index 0);
180             usuario.setEmail(jsonObject.optString( name: "Email"));
181             usuario.setPassword(etContra.getText().toString());
182             usuario.setNombre(jsonObject.optString( name: "Nombre Usuario"));
183             usuario.setId(jsonObject.optString( name: "IdUsuario"));
184             usuario.setFoto(jsonObject.optString( name: "Foto Usuario"));
185             usuario.setTelefono(jsonObject.optString( name: "Telefono Usuario"));
186             usuario.setDomicilio(jsonObject.optString( name: "Domicilio Usuario"));
187         } catch (JSONException e) {
188             e.printStackTrace();
189         }
190
191         correo = etEmail.getText().toString();
192         contra = etContra.getText().toString();
193         if (chkRecuerdame.isChecked()) {
194             loginPrefsEditor.putBoolean("guardarIngreso", true);
195             loginPrefsEditor.putString("correo", correo);
196             loginPrefsEditor.putString("contra", contra);
197             loginPrefsEditor.commit();
198         }
199     }
200 }

```

Figura 125. Uso de Volley

## 9.2 Diseño responsivo móvil

Los diseños se realizaron con ayuda del software *Adobe XD*. Como se puede ver en la Figura 126. En ese software se había diseñado un dashboard (menú principal) con dos columnas, pero a la hora de pasarlo a *Android Studio* y realizar pruebas en dispositivos físicos y emuladores digitales con diferentes tamaños de pantalla, se detectó que este diseño no era óptimo en pantallas chicas.

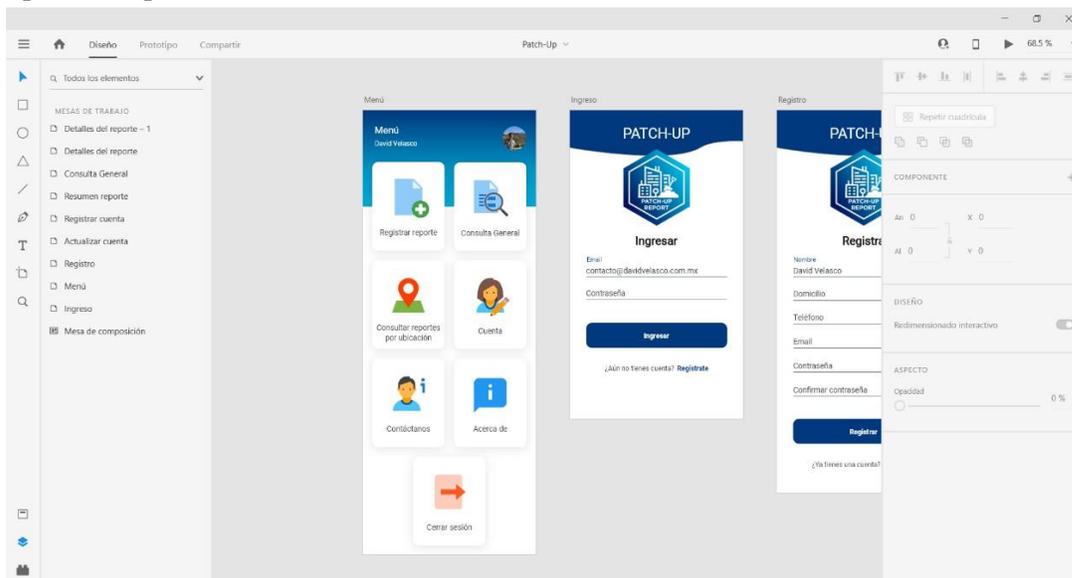


Figura 126. Uso de Adobe XD.

Para corregir estos errores en Android Studio se usó una condición de diseño, la cual se hace implementando dos Layouts, uno para el uso de pantallas con anchura mayor a 500dp y otra, para las anchuras menores. Ver en la Figura 127 el uso de estas layouts.

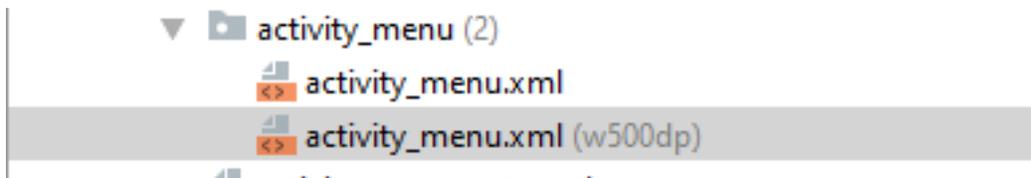


Figura 127. Dos layouts.

En la Figura 128 se presenta ver el diseño para las pantallas con anchura de pantalla mayor a 500dp, en la cual el dashboard (menú principal) consta de dos columnas, tal como se diseñó en Adobe XD. En este ejemplo se puede ver desde un emulador de un dispositivo Nexus con medidas de pantalla **800x1280dpi**, es decir, uno de 7 pulgadas.

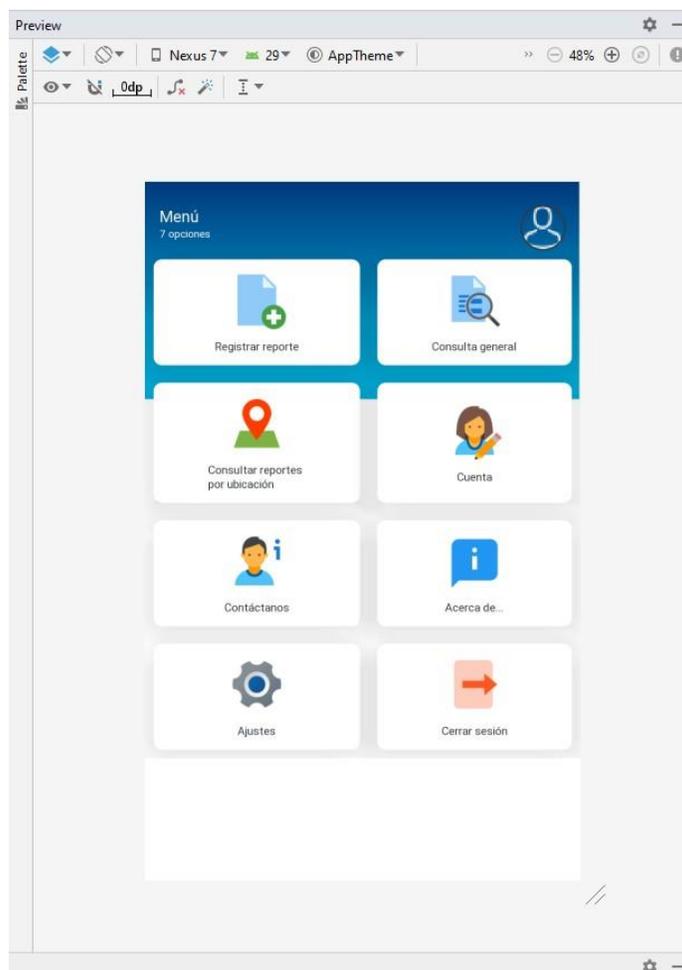
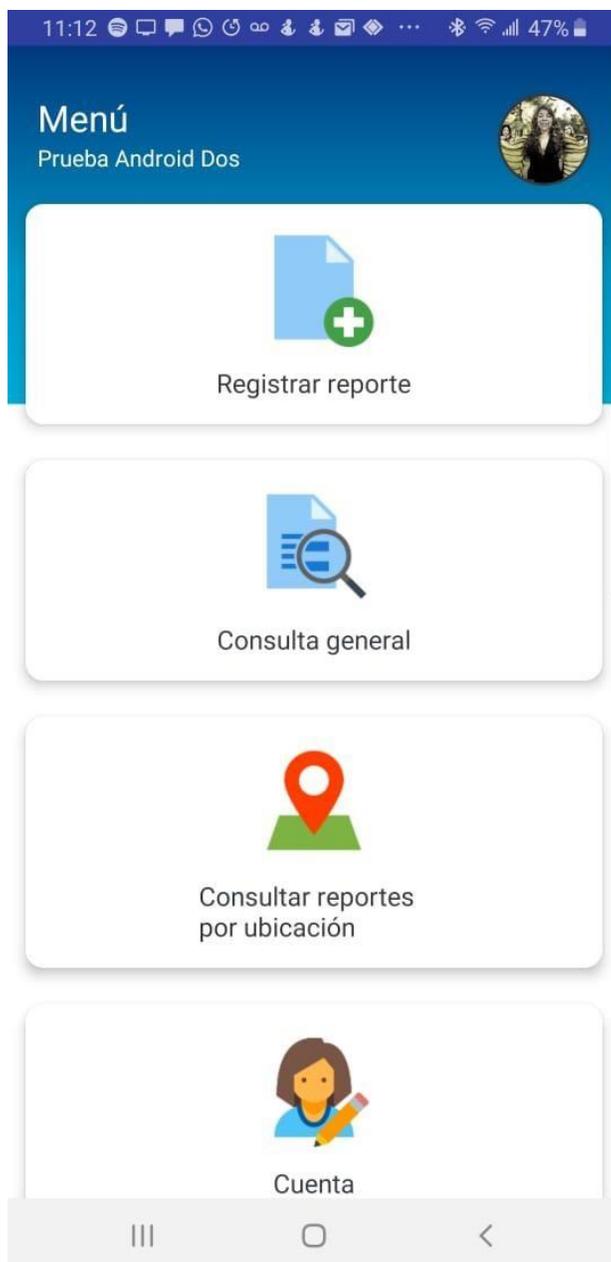


Figura 128. Diseño para pantallas con anchura mayor a 500dp.

Para las pantallas pequeñas, con anchura mayor a la mencionada, se optó por dejar una sola columna por renglón que abarca el ancho total de la pantalla. Esto se puede observar en la Figura 129, el cual es un dispositivo físico con una pantalla de 5 pulgadas.



*Figura 129. Menú para pantallas pequeñas.*

Asimismo, los diseños funcionan de la misma manera, cuando se coloca el teléfono en posición horizontal o vertical, se cambia el diseño dependiendo de la nueva anchura.

### 9.3 Imágenes desde servidor sin ayuda de librerías

Al principio, antes de hacer uso de la librería Glide, en las primeras pruebas del prototipo, se usó la manera primitiva de traer imágenes desde el servidor a la aplicación a través del servicio Web; ahí con ayuda de JSON, se convertía la imagen a formato base64, y una vez consultado el servicio Web, se hacía la conversión de nuevo, pero ahora se pasaba del formato base64 a imagen. Por ejemplo, en la Figura 130 se presenta un fragmento del código de una imagen en ese formato.

```
Fragmento del código de una imagen en formato base64

data:image/png;base64,iVBORw0KGgoAAAANSUUhEUgAAB4AAAAOqCAIAAADyuN/SAAA
AAXNSR0IArs4c6QAAAAARnQU1BAACxjww8YQUAAAAJcEhZcwAADsMAAA7DAcdvqGQA
AP+ISURBVHhe7L0JfBRV3v3dsz2zP446isuogzNsyg7iColKokBYg4DIrloAsqqsskSEAwYlb
lmrAkKBAEDghD2fRcSQEEQUGi4ADvjPCM/vu9Vb/bnV6Srrp1K8m9lfP9nM/x2Lnppbq6+
9ZJcdt3X62HIAiCIAiCIAiCIAiCIAiCIMh1+coDAAAAAAAAAAAAAAAAAAAIWA+r+ZDdSElgiAlgi
AlgiAlgiAlgiAoti6JgwlagiAlgiAlgiAlgiAlgiAlshYviUVAQAQ1BEARBEARBEARBEARBEARZi5f
KlqCAhiAlgiAlgiAlgiAlgiAlgqzFS2URUEBDEARBEARBEARBEARBEARBEARB1uKlsggooCElgiA
lgiAlgiAlgiAlgiBr8VJZBBTQEARBEARBEARBEARBEARBEARBkLV4qSwCCmgIgiAlgiAlgiAlgiAlg
iDIWrxUjmL+/Pk8RYECGolgCllgCllgCllgCllgCllWL5XDmR+A/384KKAhCllgCllgCllgCllgC
llga/FSOQTePQfgl4aAAhqCIAiCIAiCIAiCIAiCIAiCIAiyFi+VA/DWORz+swAooCElgiAlgiAlgiAlgiA
lgiBr8VJZhmIpoB+ud1/dhg/1GHbL1M03T9/2wMvDa9Z98r6HH4scBkEQBEEQBEEQBEEQ
BEEQBGGkiXiqL4HIBfd/Dj91X7+labV/8x5vzfzNp
```

Figura 130. Imagen en base64.

En la Figura 131 se presenta la misma imagen de la Figura 130, pero ahora se puede ver de forma normal, es decir, no está en formato base64.

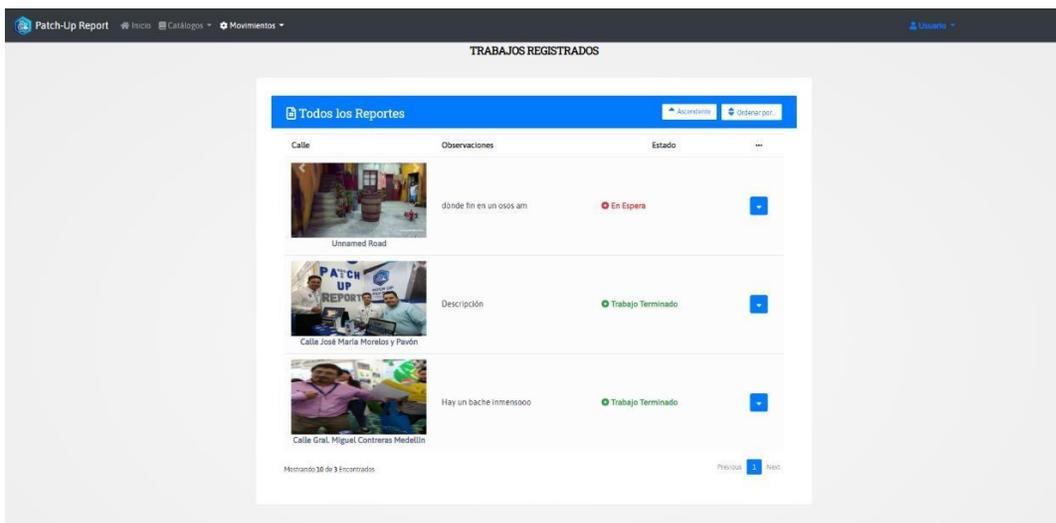


Figura 131. Imagen sin base64.

Para solucionar este problema, que a su vez ocasionaba un mayor tráfico de datos por la red, se usó la librería Glide, a la cual, para hacer una consulta de imagen alojada en algún servidor remoto, solo se le proporciona la dirección y el lugar donde se mostrará dicha imagen traída. Ver en el ejemplo de la Figura 132, la cual consulta al servidor la imagen del reporte y la muestra en un *ImageView*.

```
Reporte reporte = reporteLista.get(position);

Glide.with(mCtx)
    .load(reporte.getRutaImagen())
    .into(holder.idImagen);
```

Figura 132. Mostrando imagen con librería Glide.

## Capítulo 10. Conclusiones

Con el desarrollo de la presente tesis, se logró el objetivo de realizar un sistema multiplataforma (Web y móvil, en este caso), la cual permite gestionar de manera eficiente y eficaz los reportes realizados por usuarios, así como las acciones de mantenimiento en ciudades, empresas o instituciones públicas o privadas.

Además, uno de los principales objetivos es que mejore la infraestructura pública y privada dependiendo donde se implemente el proyecto, conservando las inversiones en construcciones y equipos disminuyendo además los riesgos por accidentes.

El presente proyecto fue aprobado por parte del H. Ayuntamiento de Zapotlán el Grande, así como también se logró integrar a pequeños industriales de la región para su implementación en un futuro inmediato, ya que se valoró su eficiencia y funcionalidad a través de varias pruebas. Por el momento se encuentra en una etapa de negociación para su implementación.

Con el desarrollo de este proyecto se reforzaron habilidades y conocimientos de temas vistos a lo largo del curso de la carrera de Ingeniería Informática, pero también se aprendieron nuevos frameworks y tecnologías innovadoras del mercado, así como diferentes lenguajes de programación.

## Capítulo 11. Referencias bibliográficas

- Andersson, David. "HTML5, XHTML 2 and the future of the Web". Digital Web Magazine, 10 abril, 2007.  
[http://www.digitalWeb.com/articles/html5\\_xhtml2\\_and\\_the\\_future\\_of\\_the\\_Web/](http://www.digitalWeb.com/articles/html5_xhtml2_and_the_future_of_the_Web/)
- Android. (2020). *Introducción a las animaciones*. 20-08-2020, de Android Studio Sitio Web: <https://developer.android.com/training/animation/overview?hl=es-419>
- Arias, M. Á. (2017). *Aprende Programacion Web con PHP y MySQL: 2ª Edicion*. IT campus Academy.
- Arias, Á. (2014). *Bases de Datos con MySQL: 2ª Edición*. IT Campus Academy.
- Aubry, C. (2012). *HTML5 y CSS3-Revolucione el diseño de sus sitios Web*. Ediciones ENI.
- Bouskela, M., Casseb, M., Bassi, S., De Luca, C. y Facchina, M. (2016). La ruta hacia las Smart Cities: Migrando de una gestión tradicional a la ciudad inteligente. Banco Interamericano de Desarrollo (BID) Recuperado el 29 de Diciembre de 2017 de <https://publications.iadb.org/bitstream/handle/11319/7743/La-ruta-hacia-las-smart-cities-Migrando-de-una-gestion-tradicional-a-la-ciudad-inteligente.pdf>
- Brent, K. (2014). Generación de una API de cliente JavaScript para un servicio de integración.
- Deitel, H. M., & Deitel, P. J. (2003). *Como programar en Java*. Pearson Educación.
- Firebase. (2019). *Iniciando con Firebase*. 20-08-2020, de Google Sitio Web: <https://firebase.google.com/docs/android/setup?hl=es>
- Franganillo, J. (2011). Html5: el nuevo estándar básico de la Web. *Anuario ThinkEPI*, 5(1), 261-265.
- Gauchat, J. D. (2012). *El gran libro de HTML5, CSS3 y Javascript*. Marcombo.
- Gironés, J. T. (2012). El Gran Libro de Android (Segunda edición ed.). *Barcelona, España: Grupo Editor Alfaomega*.
- COBO, Á., GÓMEZ, P., PÉREZ, D., & ROCHA, R. (2005). PHP y MySQL Tecnologías para el desarrollo de aplicaciones Web, 497.
- Keith, J., & Andrew, R. (2010). *HTML5 for Web designers* (Vol. 1). New York, NY: A Book Apart.
- Lee, D. (2017). Garmin | What is GPS? Obtenido de <https://goo.gl/LcwM5r>

Mayavsky, M. (2013). *Economía de las Nubes: La computación en la nube, desde el lejano ayer, por el profundo hoy y el distante mañana*. Grin Publishing: USA. Recuperado de <http://www.grin.com/es/e-book/313242/economía-de-las-nubes>.

Muñoz, V. J. E. (2012). *HTML, presente y futuro de la Web*. Vicente Javier Eslava Muñoz.

PhilJay. "PhilJay/MPAndroidChart." GitHub, 01 Agosto. 2020,  
[github.com/PhilJay/MPAndroidChart](https://github.com/PhilJay/MPAndroidChart)

Mariño, R. (2015). *Metodología y técnicas para la implementación de sitios Web y para el diseño gráfico*. Ideasropias Editorial.

# Anexos

## Adapter.java

```
package com.example.patch_up;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;
import com.google.android.material.snackbar.Snackbar;

import java.util.List;

public class Adapter extends RecyclerView.Adapter<Adapter.ReporteViewHolder> {

    private Context mCtx;

    private List<Reporte> reporteLista;

    public Adapter(Context mCtx, List<Reporte> reporteLista) {
        this.mCtx = mCtx;
        this.reporteLista = reporteLista;
    }

    @Override
    public ReporteViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(mCtx);
        View view = inflater.inflate(R.layout.list_layout, parent, false);
        return new ReporteViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ReporteViewHolder holder, int position) {
        Reporte reporte = reporteLista.get(position);

        Glide.with(mCtx)
            .load(reporte.getRutaImagen())
            .into(holder.idImagen);
    }
}
```

```

        holder.tvIdReporte.setText(mCtx.getString(R.string.consulta_reporte) + ": " +
reporte.getIdReporte());
        holder.tvCalle.setText(reporte.getCalle());
        holder.tvFecha.setText(mCtx.getString(R.string.consulta_fecha) + ": " +
reporte.getFecha());
        holder.tvEstado.setText(mCtx.getString(R.string.consulta_estado) + ": " +
reporte.getEstado());
        holder.tvId11.setText(reporte.getIdReporte());

    }

    @Override
    public int getItemCount() {
        return reporteLista.size();
    }

    // Provide a reference to the views for each data item
    // Complex data items may need more than one view per item, and
    // you provide access to all the views for a data item in a view holder
    class ReporteViewHolder extends RecyclerView.ViewHolder {

        TextView tvIdReporte, tvCalle, tvFecha, tvEstado, tvId11;
        ImageView idImagen;
        CardView cardView;

        public ReporteViewHolder(View itemView){
            super(itemView);

            tvIdReporte = itemView.findViewById(R.id.tvIdReporte11);
            tvCalle = itemView.findViewById(R.id.tvCalle11);
            tvFecha = itemView.findViewById(R.id.tvFecha11);
            tvEstado = itemView.findViewById(R.id.tvEstado11);
            tvId11 = itemView.findViewById(R.id.tvId11);
            idImagen = itemView.findViewById(R.id.idImagen11);
            cardView = itemView.findViewById(R.id.cardView);

            cardView.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Snackbar snackbar = Snackbar.make(cardView, "Click en: " +
tvCalle.getText().toString(), Snackbar.LENGTH_LONG);
                    snackbar.show();

                    iniciarActividad(tvId11.getText().toString());
                }
            });
        }

        public void iniciarActividad(String idReporte){
            Intent intent = new Intent(mCtx, DetallesActivity.class);
            intent.putExtra("IdReporte", idReporte);
            mCtx.startActivity(intent);
        }
    }
}

```

## activity\_resultados.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".ResultadosActivity"
android:background="@drawable/fondodos">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginStart="0dp"
        android:layout_marginTop="0dp"
        android:layout_marginEnd="0dp"
        android:layout_marginBottom="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="vertical">

            <RelativeLayout
                android:layout_marginTop="40dp"
                android:layout_marginLeft="20dp"
                android:layout_marginRight="20dp"
                android:layout_width="match_parent"
                android:layout_height="wrap_content">

                <TextView
                    android:id="@+id/tvIdReporte"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_alignWithParentIfMissing="false"
                    android:layout_centerHorizontal="true"
                    android:fontFamily="@font/roboto"
                    android:text="@string/resultados_reporte"
                    android:textColor="@color/primariotres"
                    android:textSize="24sp"
                    android:textStyle="bold"/>

            </RelativeLayout>

            <RelativeLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginLeft="20dp"
                android:layout_marginTop="20dp"
                android:layout_marginRight="20dp">

                <TextView
                    android:id="@+id/tvAntes"
                    android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_alignWithParentIfMissing="false"
        android:fontFamily="@font/roboto"
        android:text="@string/antes"
        android:textColor="@color/primariouno"
        android:textSize="12sp"/>

<ImageView
    android:id="@+id/ivAntes"
    android:layout_width="280dp"
    android:layout_height="190dp"
    android:layout_below="@id/tvAntes"
    android:src="@drawable/noimage"/>

</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="10dp"
    android:layout_marginRight="20dp">

    <TextView
        android:id="@+id/tvDespues"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignWithParentIfMissing="false"
        android:fontFamily="@font/roboto"
        android:text="@string/despues"
        android:textColor="@color/primariouno"
        android:textSize="12sp"/>

    <ImageView
        android:id="@+id/ivDespues"
        android:layout_width="280dp"
        android:layout_height="190dp"
        android:layout_below="@id/tvDespues"
        android:src="@drawable/noimage"/>

</RelativeLayout>

<GridLayout
    android:columnCount="1"
    android:rowCount="4"
    android:alignmentMode="alignMargins"
    android:columnOrderPreserved="false"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <com.google.android.material.textfield.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/etFechaReporte"

style="@style/Widget.MaterialComponents.TextInputEditText.FilledBox.Dense"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="4dp"
        android:layout_marginRight="16dp"

```

```

        android:layout_marginBottom="4dp"
        android:fontFamily="@font/roboto"
        android:hint="@string/fecha_reporte"
        android:inputType="text"
        android:textColor="@color/primariotres"
        android:textSize="16sp"
        android:enabled="false"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/etFechaTerminacion"
style="@style/Widget.MaterialComponents.TextInputEditText.FilledBox.Dense"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="4dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="4dp"
        android:fontFamily="@font/roboto"
        android:hint="@string/fecha_terminacion"
        android:inputType="text"
        android:textColor="@color/primariotres"
        android:textSize="16sp"
        android:enabled="false"/>
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/etComentarios"
style="@style/Widget.MaterialComponents.TextInputEditText.FilledBox.Dense"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="4dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="4dp"
        android:fontFamily="@font/roboto"
        android:hint="@string/comentario_empleado"
        android:inputType="textMultiLine"
        android:lines="4"
        android:maxLength="180"
        android:textColor="@color/primariotres"
        android:textSize="16sp"
        android:enabled="false"/>
</com.google.android.material.textfield.TextInputLayout>
<Button
    android:id="@+id/btVolver"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_marginLeft="16dp"

```

```

        android:layout_marginTop="28dp"
        android:layout_marginRight="16dp"
        android:layout_marginBottom="32dp"
        android:fontFamily="@font/roboto"
        android:text="@string/bt_volvercons"
        android:textStyle="bold"
        app:backgroundTint="#00397D"
        app:cornerRadius="12dp" />

    </GridLayout>

</LinearLayout>
</ScrollView>

</androidx.constraintlayout.widget.ConstraintLayout>

```

## Provedores.php

```

<br><br>
<?php

$page_title = "Registrar Proveedor";

include 'link.php';

?>

<script src="resource\js\proveedores.js"></script>

<div id="contain">
    <div class="page-header">
        <h1 align="center"><?php echo $page_title; ?></h1>
    </div>
    <div class="container">
<form name="registrarProveedor" id="registrarProveedor" method="POST">
    <div class="card col-md-12">
        <div class="card-body">
            <div class="form-row">
                <?php
                    $re=$conexion-
>query("SELECT count(*) as contador from proveedores");
                    $dato=mysqli_fetch_assoc($re);
                    $dato['contador'];
                    $Id=$dato['contador']+1;
                ?>
            </div>
        </div>
    </div>
</form>

```

```

<div class="form-group col-md-3">
    <label for="txtIdProveedor">IdProveedor:</label>
    <input type="number" name="txtIdProveedor" class="form-
control" value="<?php echo $Id; ?>" readonly required>
</div>
<div class="form-group col-md-5">
    <label for="txtNombre">Nombre:</label>
    <input type="text" name="txtNombre" class="form-
control" required>
</div>
<div class="form-group col-md-4">
    <label for="txtDomicilio">Domicilio:</label>
    <input type="text" name="txtDomicilio" class="form-
control" required>
</div>
<div class="form-group col-md-4">
    <label for="txtColonia">Colonia:</label>
    <input type="text" name="txtColonia" class="form-
control" required>
</div>
<div class="form-group col-md-4">
    <label for="txtCP">C.P:</label>
    <input type="text" name="txtCP" class="form-control" required>
</div>
<div class="form-group col-md-4">
    <label for="txtTelefono">Telefono:</label>
    <input type="text" name="txtTelefono" class="form-control">
</div>
<div class="form-group col-md-4">
    <label for="txtCelular">Celular:</label>
    <input type="text" name="txtCelular" class="form-control">
</div>
<div class="form-group col-md-4">
    <label for="txtCiudad">Ciudad:</label>
    <input type="text" name="txtCiudad" class="form-
control" required>
</div>
<div class="form-group col-md-4">
    <label for="txtEntidad">Entidad:</label>
    <input type="text" name="txtEntidad" class="form-
control" required>
</div>
<div class="form-group col-md-4">
    <label for="txtRFC">R.F.C:</label>
    <input type="text" name="txtRFC" class="form-control" required>

```

```

    </div>
    <div class="form-group col-md-4">
      <label for="txtEmail">Email:</label>
      <input type="text" name="txtEmail" class="form-
control" required>
    </div>
    <div class="form-group col-md-4">
      <label for="txtSaldo">Saldo:</label>
      <input type="text" name="txtSaldo" class="form-
control" step="0.1" required>
    </div>
    <div class="form-group col-md-4">
      <input type="button" class="form-control btn btn-
primary" id="btnCons" value="Consulta General">
    </div>
    <div class="form-group col-md-4">
      <input type="submit" name="cmdGrabar" class="form-
control btn btn-primary" value="Grabar">

    </div>
    <div class="form-group col-md-4">
      <input type="reset" value="Cancelar" class="form-
control sbtn btn-primary">
    </div>

  </div>
</div>
</div>
</div>
</div>
</form>
</div>
</div>

```