



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE INGENIERÍA EN SISTEMA COMPUTACIONALES E  
INFORMÁTICOS**

**Tema:**

---

SISTEMA WEB PARA LA GESTIÓN DE SERVICIO TÉCNICO APLICANDO  
GEORREFERENCIACIÓN PARA LA RECEPCIÓN Y ENTREGA DE  
DISPOSITIVOS TECNOLÓGICOS EN LA EMPRESA EPC COMPU.

---

**Trabajo de Titulación Modalidad:** Proyecto de Investigación, presentado previo a  
la obtención del título de Ingeniera en Sistemas Computacionales e Informáticos

**ÁREA:** Software

**LÍNEA DE INVESTIGACIÓN:** Desarrollo de Software

**AUTOR:** Valeria Alexandra Quispe Pérez

**TUTOR:** Ing. Hernando Buenaño Valencia, Mg.

**Ambato – Ecuador**

**septiembre – 2022**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del Trabajo de Titulación con el tema: SISTEMA WEB PARA LA GESTIÓN DE SERVICIO TÉCNICO APLICANDO GEORREFERENCIACIÓN PARA LA RECEPCIÓN Y ENTREGA DE DISPOSITIVOS TECNOLÓGICOS EN LA EMPRESA EPC COMPU, desarrollado bajo la modalidad Proyecto de Investigación por el señorita Valeria Alexandra Quispe Pérez, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que la estudiante ha sido tutorada durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, septiembre 2022.

-----  
Ing. Mg. Hernando Buenaño Valencia

TUTOR

## AUTORÍA

El presente Proyecto de Investigación titulado: SISTEMA WEB PARA LA GESTIÓN DE SERVICIO TÉCNICO APLICANDO GEORREFERENCIACIÓN PARA LA RECEPCIÓN Y ENTREGA DE DISPOSITIVOS TECNOLÓGICOS EN LA EMPRESA EPC COMPU, es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, septiembre 2022.



Valeria Alexandra Quispe Pérez

C.C. 1805143813

AUTOR

## **APROBACIÓN TRIBUNAL DE GRADO**

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señorita Valeria Alexandra Quispe Pérez, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado SISTEMA WEB PARA LA GESTIÓN DE SERVICIO TÉCNICO APLICANDO GEORREFERENCIACIÓN PARA LA RECEPCIÓN Y ENTREGA DE DISPOSITIVOS TECNOLÓGICOS EN LA EMPRESA EPC COMPU, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, septiembre 2022.

-----

Ing. Pilar Urrutia, Mg.

PRESIDENTE DEL TRIBUNAL

-----

Ing. Carlos Núñez

PROFESOR CALIFICADOR

-----

Ing. Julio Balarezo

PROFESOR CALIFICADOR

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, Septiembre 2022.



---

Valeria Alexandra Quispe Pérez  
C.E. 180514381-3  
AUTOR

## **DEDICATORIA**

*El presente proyecto de Investigación va dedicado en primer lugar a mis padres quienes han sido el pilar fundamental, mi centro de apoyo durante todo el largo camino que he tenido que recorrer, sin ellos nada de esto sería posible.*

*Mi hermana quien ha sido el ejemplo de superación, mi compañera de aventuras durante toda mi vida y quien me ayudo a ser una mejor persona.*

*Mis abuelitos quienes soñaban con el día en que me convierta en una gran profesional, y estoy segura que desde el cielo estarán muy orgullosos de mí.*

*A toda mi familia, quienes han estado los momentos buenos y malos, siempre recordare cada uno de sus consejos.*

***Valeria Alexandra Quispe Pérez***

## AGRADECIMIENTO

*A Dios por permitirme terminar mi carrera universitaria, ya que me ha brindado fortaleza y sabiduría a lo largo de este camino.*

*A la Facultad de Ingeniería en Sistemas, Electrónica e Industrial en la Universidad Técnica de Ambato, por brindarme la oportunidad de formarme como profesional con los mejores docentes, quienes han impartido sus conocimientos, han pulido cada fortaleza de sus estudiantes y me han brindado las mejores experiencias durante toda mi vida universitaria.*

*A EPC Compu en especial a su gerente general Ricardo Perez por abrirme las puertas de su empresa y apoyarme incondicionalmente en mi proyecto de investigación.*

*Finalmente, aquellos amigos que se convirtieron en familia, mil gracias por ese apoyo incondicional en mi vida personal y profesional.*

**Valeria Alexandra Quispe Pérez**

## INDICE

ÍNDICE DE FIGURAS .....	x
RESUMEN EJECUTIVO .....	xx
ABSTRACT .....	xxi
CAPÍTULO I.- MARCO TEÓRICO .....	22
1.1 Tema de Investigación .....	22
1.2 Antecedentes Investigativos.....	22
1.2.1 Contextualización del Problema.....	24
1.2.2 Delimitación.....	26
1.2.3 Justificación.....	26
1.2.4 Fundamentación Teórica .....	27
1.3 Objetivos .....	40
1.3.1 Objetivo General .....	40
1.3.2 Objetivos Específicos.....	40
2. CAPÍTULO II.- METODOLOGÍA .....	41
2.1 Materiales .....	41
2.2 Métodos .....	45
2.2.1 Modalidad de investigación .....	45
2.2.2 Población y muestra .....	46
2.2.3 Recolección de la información.....	47
2.2.4 Procesamiento y Análisis de Datos .....	63
2.2.4.1 Análisis e Interpretación de Resultados de la Encuesta Aplicada.....	63
CAPÍTULO III .....	65
RESULTADOS Y DISCUSIONES .....	65
3.1 Análisis e Interpretación de resultados.....	65
3.1.1 Descripción del Proceso de Soporte Técnico en la Empresa EPC Compu	65
3.1.2 Estudio Comparativo de Tecnologías de Front-End .....	66
3.1.3 Estudio Comparativo de Tecnologías de Back-End.....	69
3.1.4 Metodología para el Desarrollo del Software .....	73
3.2 Desarrollo de la Propuesta.....	80
3.3 Aplicación de la Metodología .....	80
3.3.1 Arquitectura de la Aplicación .....	81



3.3.2 Visualizar el flujo de Trabajo.....	82
3.3.3 Limitar la cantidad de trabajo (WIP).....	93
3.3.4 Flujo de Trabajo .....	95
3.3.5 Políticas Explicitas .....	96
3.3.6 Desarrollo de Tareas.....	96
3.3.7 Retroalimentación .....	250
CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES .....	250
4.1 Conclusiones .....	250
4.2 Recomendaciones.....	251
BIBLIOGRAFÍA.....	253
ANEXOS.....	255

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Representación sistema de posicionamiento global .....	29
<b>Figura 2.</b> Precios de Api de Geolocalización en función de los mapas .....	31
<b>Figura 3.</b> Precios de Api de Geolocalización en función de las rutas .....	31
<b>Figura 4.</b> Precios de Api de Geolocalización en función de las rutas .....	31
<b>Figura 5.</b> Precios de Api de Geolocalización en función de los sitios .....	31
<b>Figura 6.</b> Precios de Api de Geolocalización en función de los sitios [33].....	31
<b>Figura 7.</b> Frecuencia con la que se utiliza un dispositivo tecnológico .....	49
<b>Figura 8.</b> Tipos de dispositivos tecnológico utilizados en la vida cotidiana .....	50
<b>Figura 9.</b> Atención por parte de soporte técnico en cuanto a incidentes.....	51
<b>Figura 10.</b> Eficiencia en el proceso de recepción y entrega de los dispositivos tecnológicos.....	52
<b>Figura 11.</b> Medios por los cuales soporte técnico obtienen información.....	53
<b>Figura 12.</b> Aceptación de herramientas tecnológicas para la mejora de procesos de entrega-recepción de dispositivos. ....	54
<b>Figura 13.</b> Lista de dispositivos informáticos para obtener información en el proceso de soporte técnico.....	55
<b>Figura 14.</b> Afirmación sobre la información sobre el estado de reparación del dispositivo tecnológico.....	56
<b>Figura 15.</b> Descripción del Proceso de Soporte Técnico en la Empresa EPC Compu .....	65
<b>Figura 16.</b> Diagrama de Flujo del Sistema- Usuario Motorizado .....	91
<b>Figura 17.</b> Diagrama de Flujo del Sistema- Usuario Técnico.....	92
<b>Figura 18.</b> Diagrama de Flujo del Sistema- Usuario Secretaria.....	92
<b>Figura 19.</b> Diagrama de Flujo del Sistema- Usuario.....	93
<b>Figura 20.</b> Creación de proyecto herramienta Trello para Kanban.....	94
<b>Figura 21.</b> Lista de tareas a realizar en el presente proyecto.....	95
<b>Figura 22.</b> Flujo de trabajo metodología Kanban.....	95
<b>Figura 23.</b> Subtareas por realizar partiendo de la metodología Kanban .....	96
<b>Figura 24.</b> Políticas explícitas, parte de la metodología Kanban .....	96
<b>Figura 25.</b> Tareas y Subtareas del Diseño de la Base de Datos .....	97
<b>Figura 26.</b> Entidades de la Base de Datos .....	97

<b>Figura 27.</b> Relaciones de la Base de Datos .....	98
<b>Figura 28.</b> Pruebas del Diseño de la Base de Datos .....	98
<b>Figura 29.</b> Modelo Entidad-Relación de la Base de Datos .....	99
<b>Figura 30.</b> Arquitectura de Capas del Sistema .....	100
<b>Figura 31.</b> Conexión a la Base de Datos .....	100
<b>Figura 32.</b> Acceso a la Base de Datos .....	100
<b>Figura 33.</b> Diseño de la Base de Datos Completada .....	101
<b>Figura 34.</b> Prueba del Diseño de la Base de Datos Completada .....	101
<b>Figura 35.</b> Tareas y Subtareas del Diseño de la página principal de presnetación .	102
<b>Figura 36.</b> Pruebas del Diseño de la página principal de presentación.....	103
<b>Figura 37.</b> Página principal de Sistema EPC-Compu .....	103
<b>Figura 38.</b> Secciones de la página principal del Sistema web.....	104
<b>Figura 39.</b> Recursos Bootstrap en la capa de interfaz .....	105
<b>Figura 40.</b> Referencias de los recursos de Bootstrap .....	105
<b>Figura 41.</b> Navegación de secciones, página principal .....	106
<b>Figura 42.</b> Diseño de la página principal de presentación Completada .....	106
<b>Figura 43.</b> Pruebas del Diseño de la Página Principal de presentación.....	107
<b>Figura 44.</b> Página de creación de nuevos usuarios.....	108
<b>Figura 45.</b> Pruebas de la Página de creación de nuevos usuarios .....	109
<b>Figura 46.</b> Formulario de registro de nuevo usuario .....	109
<b>Figura 47.</b> Método de encriptación .....	110
<b>Figura 48.</b> Algoritmo de encriptación .....	110
<b>Figura 49.</b> Método registra nuevo usuario .....	111
<b>Figura 50.</b> Método de guardar persona en la capa de datos .....	112
<b>Figura 51.</b> Método de guardar usuario en la capa de datos.....	112
<b>Figura 52.</b> Control de excepción registro de nuevo usuario.....	113
<b>Figura 53.</b> Página de creación de nuevos usuarios completada .....	113
<b>Figura 54.</b> Pruebas de la página de creación de nuevos usuarios completada .....	114
<b>Figura 55.</b> Acceso al Sistema .....	115
<b>Figura 56.</b> Pruebas de Acceso al sistema .....	116
<b>Figura 57.</b> Formulario de acceso al sistema .....	116
<b>Figura 58.</b> Método de desencriptación .....	117
<b>Figura 59.</b> Algoritmo de desencriptación.....	117

<b>Figura 60.</b> Método de validar ingreso en la capa de interfaz .....	118
<b>Figura 61.</b> Método de obtener usuario en la capa de negocio .....	118
<b>Figura 62.</b> Método de obtener usuario en la capa de datos .....	119
<b>Figura 63.</b> Acceso al Sistema Completada.....	119
<b>Figura 64.</b> Pruebas de Acceso al Sistema completada .....	120
<b>Figura 65.</b> Página principal de acceso al sistema .....	121
<b>Figura 66.</b> Pruebas de la página principal de acceso al sistema.....	122
<b>Figura 67.</b> Página principal de acceso al sistema .....	122
<b>Figura 68.</b> Diseño y secciones de menú de la página de inicio.....	123
<b>Figura 69.</b> Página principal de acceso al sistema completada.....	123
<b>Figura 70.</b> Pruebas de la página principal de acceso al sistema completada .....	124
<b>Figura 71.</b> Gestión de Retiros.....	125
<b>Figura 72.</b> Subtareas de la historia de usuario HU06 Solicitar Retiro .....	126
<b>Figura 73.</b> Subtarea de la historia de usuario HU06 - Estado de Retiros.....	127
<b>Figura 74.</b> Sub-Tareas de la Historia de Usuario HU06 - Asignar Retiro .....	128
<b>Figura 75.</b> Sub-Tarea de la Historia de Usuario HU06 - Ver Asignación .....	129
<b>Figura 76.</b> Sub-Tarea de la Historia de Usuario HU06 - Ingresar Retiro.....	130
<b>Figura 77.</b> Pruebas de la Gestión de Retiros .....	131
<b>Figura 78.</b> Formulario de solicitar retiro .....	132
<b>Figura 79.</b> Uso de la Api de Google Maps.....	132
<b>Figura 80.</b> Llave de la Api de Google Maps .....	133
<b>Figura 81. Sección de no ser agregado correctamente</b> .....	133
<b>Figura 82.</b> Sección de ser agregado correctamente.....	134
<b>Figura 83.</b> Método de validad retiro en la capa de interfaz.....	135
<b>Figura 84.</b> Método de guardar retiro en la capa de interfaz .....	136
<b>Figura 85.</b> Método de guardar retiro de la capa de negocio .....	137
<b>Figura 86.</b> Método de guardar retiro de la capa de datos .....	138
<b>Figura 87.</b> Formulario de visualizar estado de retiro .....	138
<b>Figura 88.</b> Informe de historial de estado de un retiro .....	139
<b>Figura 89.</b> Método de recuperar retiros en la capa de interfaz.....	140
<b>Figura 90.</b> Método de recuperar retiros por usuario en la capa de datos.....	141
<b>Figura 91.</b> Método de generar informe en la capa de interfaz.....	142
<b>Figura 92.</b> Vista de asignaciones de retiros pendientes.....	143

<b>Figura 93.</b> Formulario de registrar asignación de retiros .....	143
<b>Figura 94.</b> Vista de retiros solicitados pendientes.....	144
<b>Figura 95.</b> Vista de personal de entrega y recepción.....	144
<b>Figura 96.</b> Método de visualizar asignaciones de retiros en la capa de interfaz .....	145
<b>Figura 97. Llamado a métodos de la capa de negocio</b> .....	145
<b>Figura 98.</b> Método de obtener asignaciones de retiros pendientes en la capa de negocio .....	146
<b>Figura 99.</b> Método de obtener asignaciones de retiros pendientes en la capa de datos .....	146
<b>Figura 100.</b> Método de guardar asignación de retiro en la capa de interfaz .....	147
<b>Figura 101.</b> Método de guardar asignación de retiro en la capa de datos .....	148
<b>Figura 102.</b> Método de retiros pendientes en la capa de interfaz.....	148
<b>Figura 103.</b> Método de recuperar retiros pendientes en la capa de datos.....	149
<b>Figura 104.</b> Método de recuperar de personal de entrega y recepción en la capa de interfaz.....	150
<b>Figura 105.</b> Método de recuperar personal de entrega y recepción en la capa de datos .....	151
<b>Figura 106.</b> Vista de retiros asignados al personal de entrega y recepción.....	152
<b>Figura 107.</b> Visualización de Asignación .....	152
<b>Figura 108.</b> Visualización de detalle de asignación de retiro.....	153
<b>Figura 109.</b> Método de obtener asignaciones de retiros por usuario pendientes en la capa de interfaz .....	153
<b>Figura 110.</b> Método de obtener asignaciones de retiros por usuario pendientes en la capa de datos .....	154
<b>Figura 111.</b> Método para enviar datos de retiro .....	154
<b>Figura 112.</b> Método de recuperar retiro por id en la capa de interfaz.....	155
<b>Figura 113.</b> Método de recuperar coordenada geográfica en la capa de datos.....	155
<b>Figura 114.</b> Método de trazado de ruta en la capa de interfaz .....	156
<b>Figura 115.</b> Vista de listado de ingresos de retiros .....	156
<b>Figura 116.</b> Formulario de ingresar retiro .....	157
<b>Figura 117.</b> Vista de retiros asignados-Pendientes.....	158
<b>Figura 118.</b> Método cargar listado de ingresos no completados en la capa de datos .....	159

<b>Figura 119.</b> Método de cargar listado de asignaciones de retiro en la capa de interfaz .....	160
<b>Figura 120.</b> Método de registro de ingreso de retiro en la capa de interfaz .....	161
<b>Figura 121.</b> Método para guardar ingreso de retiro.....	162
<b>Figura 122.</b> Tareas y Sub-Tareas de Gestión de Retiros.....	163
<b>Figura 123.</b> Pruebas de la Gestión de Retiros .....	164
<b>Figura 124.</b> Menú de Retiros.....	165
<b>Figura 125.</b> Pruebas del Menú de Retiros .....	166
<b>Figura 126.</b> Opciones del menú de retiro .....	167
<b>Figura 127.</b> Redirección de formularios de gestión de retiros .....	167
<b>Figura 128.</b> Habilitación de operaciones de gestión de retiros por perfil .....	168
<b>Figura 129.</b> Menú de Retiros completada .....	168
<b>Figura 130.</b> Menú de Retiros completada .....	169
<b>Figura 131.</b> Gestión de Diagnósticos .....	170
<b>Figura 132.</b> Sub-Tarea de la Historia de Usuario HU08 - Seleccionar Ingreso .....	171
<b>Figura 133.</b> Sub-Tarea de la Historia de Usuario HU08 - Ingresar Diagnóstico ....	172
<b>Figura 134.</b> Sub-Tarea de la Historia de Usuario HU08 - Aceptar Diagnóstico.....	173
<b>Figura 135.</b> Sub-Tarea de la Historia de Usuario HU08 - Ver Diagnósticos.....	174
<b>Figura 136.</b> Prueba de la Gestión de Diagnósticos .....	175
<b>Figura 137.</b> Vista de ingresos disponibles sin diagnóstico .....	175
<b>Figura 138.</b> Formulario de registro de selección de ingreso .....	176
<b>Figura 139.</b> Método de recuperación de ingresos disponibles en la capa de interfaz .....	177
<b>Figura 140.</b> Método de recuperar ingresos disponibles en la capa de datos .....	178
<b>Figura 141.</b> Método de recuperar información del ingreso en la capa de interfaz..	178
<b>Figura 142.</b> Método de control de tiempo estimado selección de ingreso .....	179
<b>Figura 143.</b> Método de guardar selección de ingreso en la capa de interfaz.....	180
<b>Figura 144.</b> Vista de ingresos seleccionados por personal técnico .....	180
<b>Figura 145.</b> Formulario de ingreso de diagnóstico.....	181
<b>Figura 146.</b> Vista emergente de dispositivos a diagnosticar .....	182
<b>Figura 147.</b> Método de obtener selecciones de ingreso por usuario en la capa de interfaz.....	182
<b>Figura 148.</b> Método de recuperar información de la selección de ingreso .....	183

<b>Figura 149.</b> Método de agregar revisión de dispositivo .....	184
<b>Figura 150.</b> Método de cálculo de valor total.....	184
<b>Figura 151.</b> Método de registro de diagnóstico en la capa de interfaz.....	185
<b>Figura 152.</b> Método de registro de diagnóstico en la capa de datos.....	186
<b>Figura 153.</b> Vista de listado de diagnóstico perteneciente a un cliente.....	187
<b>Figura 154.</b> Informe de diagnóstico .....	188
<b>Figura 155.</b> Ventanas emergentes de aceptación de diagnóstico .....	188
<b>Figura 156.</b> Ventanas emergentes de rechazo de diagnóstico.....	189
<b>Figura 157.</b> Método de obtener diagnósticos por usuario en la capa de interfaz ....	189
<b>Figura 158.</b> Método de generación de informe de diagnóstico.....	191
<b>Figura 159.</b> Método de aceptación del diagnóstico en la capa de interfaz.....	191
<b>Figura 160.</b> Método de rechazo de diagnóstico en la capa de interfaz.....	192
<b>Figura 161.</b> Visualización de Diagnósticos.....	192
<b>Figura 162.</b> Método de recuperar diagnósticos por usuario en la capa de interfaz.	193
<b>Figura 163.</b> Gestión de Diagnósticos completada.....	193
<b>Figura 164.</b> Pruebas de Gestión de Diagnósticos completada .....	194
<b>Figura 165.</b> Menú de Diagnósticos .....	195
<b>Figura 166.</b> Pruebas del Menú de Diagnósticos.....	196
<b>Figura 167.</b> Menú de Diagnósticos .....	197
<b>Figura 168.</b> Redirección de formularios de gestión de diagnósticos .....	197
<b>Figura 169.</b> Habilitación de operación de gestión de diagnósticos por perfil.....	198
<b>Figura 170.</b> Menú de Diagnósticos completada.....	198
<b>Figura 171.</b> Menú de Diagnósticos completada.....	199
<b>Figura 172.</b> Gestión de Reparación - Mantenimiento .....	200
<b>Figura 173.</b> Sub-Tarea de la Historia de Usuario HU10 - Selección de Diagnóstico .....	201
<b>Figura 174.</b> Sub-Tarea de la Historia de Usuario HU10 - Finalización de Reparación/Mantenimiento .....	202
<b>Figura 175.</b> Sub-Tarea de la Historia de Usuario HU10 - Ver estado de reparación/mantenimiento.....	203
<b>Figura 176.</b> Sub-Tarea de Historia de Usuario HU10 - Completar pago/entrega...	204
<b>Figura 177.</b> Pruebas de Gestión de Reparación/Mantenimiento .....	205
<b>Figura 178.</b> Vista de diagnóstico aceptados y disponibles.....	205

<b>Figura 179.</b> Formulario de registro de selección de diagnóstico.....	206
<b>Figura 180.</b> Método de obtener diagnósticos aceptados libres en la capa de interfaz .....	206
<b>Figura 181.</b> Método de recuperar datos de diagnóstico en la capa de interfaz .....	207
<b>Figura 182.</b> Método de guardar selección de diagnóstico en la capa de interfaz....	208
<b>Figura 183.</b> Vista de diagnósticos seleccionados por usuario.....	208
<b>Figura 184.</b> Formulario de finalización de reparación .....	209
<b>Figura 185.</b> Método de obtener reparaciones iniciado por usuario en la capa de interfaz.....	209
<b>Figura 186.</b> Método de cargar reparación en la capa de interfaz .....	210
<b>Figura 187.</b> Método de finalizar reparación en la capa de interfaz.....	211
<b>Figura 188.</b> Vista de reparaciones en proceso.....	211
<b>Figura 189.</b> Informe de estado de reparación.....	212
<b>Figura 190.</b> Método de obtención de reparaciones en proceso por usuario en la capa de interfaz.....	213
<b>Figura 191.</b> Apartado de obtención de información de reparación en la capa de interfaz.....	213
<b>Figura 192.</b> Gestión de Reparación-Mantenimiento completada.....	214
<b>Figura 193.</b> Gestión de Reparación/Mantenimiento terminada .....	215
<b>Figura 194.</b> Menú de Reparación-Mantenimiento .....	216
<b>Figura 195.</b> Pruebas para el Menú de Reparación-Mantenimiento.....	217
<b>Figura 196.</b> Menú de Reparación/Mantenimiento en el sistema.....	218
<b>Figura 197.</b> Redirección de formularios de gestión de reparación.....	218
<b>Figura 198.</b> Menú de Reparación - Mantenimiento completada.....	219
<b>Figura 199.</b> Menú de Reparación - Mantenimiento completada.....	220
<b>Figura 200.</b> Gestión de Reportes.....	221
<b>Figura 201.</b> Pruebas para la Gestión de Reportes.....	222
<b>Figura 202.</b> Mapa de calor de clientes.....	223
<b>Figura 203.</b> Diseño de presentación de opciones de informes .....	224
<b>Figura 204.</b> Informe de reparaciones totales .....	224
<b>Figura 205.</b> Método de recuperar coordenadas geográficas de clientes en la capa de datos .....	225
<b>Figura 206.</b> Método de visualización de mapa de calor en la capa de interfaz.....	226



<b>Figura 207.</b> Método de generación de informe reparaciones completadas en la capa de interfaz.....	227
<b>Figura 208.</b> Gestión de Reportes completada.....	227
<b>Figura 209.</b> Gestión de Reportes completada.....	228
<b>Figura 210.</b> Gestión de Usuarios.....	229
<b>Figura 211.</b> Pruebas para la Gestión de Usuarios.....	230
<b>Figura 212.</b> Visualización de usuarios .....	231
<b>Figura 213.</b> Formulario de registro de usuario.....	231
<b>Figura 214.</b> Formulario de actualización de usuario .....	232
<b>Figura 215.</b> Método de recuperar usuarios activos en la capa de interfaz .....	232
<b>Figura 216.</b> Método de recuperar perfiles existentes en la capa de interfaz .....	233
<b>Figura 217.</b> Método de validación de correo y datos en la capa de interfaz .....	233
<b>Figura 218.</b> Método de guardar usuario en la capa de interfaz .....	234
<b>Figura 219.</b> Método de actualizar usuario en la capa de interfaz.....	235
<b>Figura 220.</b> Método de dar baja un usuario en la capa de interfaz.....	236
<b>Figura 221.</b> Gestión de Usuarios completada.....	236
<b>Figura 222.</b> Gestión de Usuarios completada.....	237
<b>Figura 223.</b> Acceso al Menú por tipo de usuario .....	238
<b>Figura 224.</b> Pruebas para el Acceso al menú por tipo de usuario .....	239
<b>Figura 225.</b> Menú de operaciones .....	240
<b>Figura 226.</b> Habilitación de menús por perfil de usuario.....	241
<b>Figura 227.</b> Acceso al menú por tipo de usuario completada .....	241
<b>Figura 228.</b> Acceso al menú por tipo de usuario completada .....	242
<b>Figura 229.</b> Manejo de Perfil Personal de los Usuarios .....	243
<b>Figura 230.</b> Pruebas del Manejo de Perfil Personal de los Usuarios .....	244
<b>Figura 231.</b> Vista de perfil de usuario .....	245
<b>Figura 232.</b> Formulario de actualización de perfil de usuario.....	246
<b>Figura 233.</b> Método de cargar información de usuario en la capa de interfaz .....	247
<b>Figura 234.</b> Método de actualizar usuario en la capa de interfaz.....	248
<b>Figura 235.</b> Manejo de Perfil Personal de los Usuarios completada.....	249
<b>Figura 236.</b> Manejo de Perfil Personal de los Usuarios completada.....	249

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Definición de la población de Estudio .....	46
<b>Tabla 2.</b> Definición de la muestra a Evaluar .....	47
<b>Tabla 3.</b> Recolección de la información .....	47
<b>Tabla 4.</b> Frecuencia con la que se utiliza un dispositivo tecnológico .....	48
<b>Tabla 5.</b> Tipos de dispositivos tecnológico utilizados en la vida cotidiana.....	50
<b>Tabla 6.</b> Atención por parte de soporte técnico en cuanto a incidentes .....	51
<b>Tabla 7.</b> Eficiencia en el proceso de recepción y entrega de los dispositivos tecnológicos.....	52
<b>Tabla 8.</b> Medios por los cuales soporte técnico obtienen información .....	53
<b>Tabla 9.</b> Aceptación de herramientas tecnológicas para la mejora de procesos de entrega-recepción de .....	54
<b>Tabla 10.</b> Lista de dispositivos informáticos para obtener información en el proceso de soporte técnico.....	55
<b>Tabla 11.</b> Afirmación sobre la información sobre el estado de reparación del dispositivo tecnológico.....	56
<b>Tabla 12.</b> Ventajas y Desventajas de .NET Framework.....	66
<b>Tabla 13.</b> Ventajas y Desventajas de React.....	67
<b>Tabla 14.</b> Ventajas y Desventajas de Angular.....	68
<b>Tabla 15.</b> Cuadro Comparativo de Tecnologías Front-End.....	69
<b>Tabla 16.</b> Ventajas y Desventajas de PHP .....	70
<b>Tabla 17.</b> Ventajas y Desventajas de Python .....	71
<b>Tabla 18.</b> Ventajas y Desventajas de ASP.NET.....	71
<b>Tabla 19.</b> Cuadro Comparativo de Tecnologías Back-End.....	72
<b>Tabla 20.</b> Ventajas y Desventajas de Waterfall o Cascada .....	74
<b>Tabla 21.</b> Ventajas y Desventajas de Scrum .....	76
<b>Tabla 22.</b> Ventajas y Desventajas de Kanban .....	78
<b>Tabla 23.</b> Cuadro Comparativo de Metodologías de Desarrollo.....	79
<b>Tabla 24.</b> Diseño de la base de Datos.....	82
<b>Tabla 25.</b> Diseño de la página principal del Sistema de Gestión de Soporte Técnico .....	83
<b>Tabla 26.</b> Creación de nuevos usuarios .....	83
<b>Tabla 27.</b> Login del sistema .....	84

<b>Tabla 28.</b> Página principal de Ingreso al Sistema .....	84
<b>Tabla 29.</b> Gestión de Retiros .....	85
<b>Tabla 30.</b> Menú de Retiros .....	86
<b>Tabla 31.</b> Gestión de Diagnósticos.....	87
<b>Tabla 32.</b> Menú de Diagnósticos.....	88
<b>Tabla 33.</b> Gestión de Reparación – Mantenimiento .....	88
<b>Tabla 34.</b> Menú de Reparación-Mantenimiento.....	89
<b>Tabla 35.</b> Gestión de Reportes .....	89
<b>Tabla 36.</b> Gestión de Usuarios .....	90
<b>Tabla 37.</b> Acceso al menú por tipo de usuario .....	90
<b>Tabla 38.</b> Manejo del Perfil.....	91

## RESUMEN EJECUTIVO

Las PYMES durante la pandemia han ido evolucionando de manera lenta debido a los altos costos de implementación de sistemas por ello; el presente proyecto de investigación trata sobre la realización de un sistema web, para la gestión de servicio técnico aplicando georreferenciación para la recepción y entrega de dispositivos tecnológicos en la empresa EPC Compu. Permitiendo desarrollar de mejor manera su trabajo diario, mediante la innovación de tecnologías, ofreciendo un servicio eficiente al cliente. Incluso que la empresa tenga un control interno de todo lo que se está realizando.

Durante todo el proceso se utilizará varias herramientas para recabar información, entre ellas entrevistas al personal y encuestas a clientes. Toda esta información será analizada y procesada para definir cada requerimiento que la empresa necesite que se implemente dentro del sistema web.

Finalmente se realizará todas las pruebas necesarias para la verificación del sistema, una vez terminadas dichas pruebas se procederá a la implantación del sistema web en la empresa EPC Compu. Se presentará su respectivo manual de usuario y una capacitación al personal.

**Palabras clave:** ASP .NET, Kanban, sistema web, georreferenciación, gestión de servicio técnico, .NET Framework, SQL

## ABSTRACT

SMEs during the pandemic have been evolving slowly due to the high costs of implementing systems for it; This research project deals with the realization of a web system for the management of technical service applying georeferencing for the reception and delivery of technological devices in the EPC Compu company. Allowing you to better develop your daily work, through technology innovation, offering an efficient customer service. Even that the company has an internal control of everything that is being done.

Throughout the process, various tools will be used to gather information, including staff interviews and customer surveys. All this information will be analyzed and processed to define each requirement that the company needs to be implemented within the web system.

Finally, all the necessary tests will be carried out for the verification of the system, once these tests have been completed, the web system will be implemented in the EPC Compu company. Their respective user manual and training will be presented to the staff.

**Keywords:** ASP .NET, Kanban, web system, georeferencing, technical services management, .NET Framework, SQL

## **CAPÍTULO I.- MARCO TEÓRICO**

### **1.1 Tema de Investigación**

“SISTEMA WEB PARA LA GESTIÓN DEL SERVICIO TÉCNICO APLICANDO GEORREFERENCIACIÓN PARA LA RECEPCIÓN Y ENTREGA DE DISPOSITIVOS TECNOLÓGICOS EN LA EMPRESA EPC COMPU”

### **1.2 Antecedentes Investigativos**

Se realizó una búsqueda de documentación que ayude a la realización del presente proyecto mediante bibliotecas virtuales y los diferentes repositorios de universidades existentes a nivel nacional en las cuales se ha encontrado, tanto, artículos de carácter científico y proyectos de tesis que se relaciona con el tema de investigación presentado en este documento en los cuales se han encontrado los siguientes trabajos:

En el proyecto de tesis con tema “Desarrollo e implementación de un sistema web de seguimiento y evaluación de prácticas preprofesionales para la facultad de Ingeniería Escuela Civil de la PUCE”, el autor José Maldonado publicado en 2016 menciona que:

Una aplicación web es de gran importancia para las empresas debido a la practicidad de ingreso por que no necesita un Sistema Operativo compatible para que la aplicación pueda ejecutarse, al contrario, como cliente solamente necesitan un navegador web, son fáciles de actualizar nuevas versiones, sin la necesidad de la instalación de un software a todos los usuarios de la empresa.

Son de acceso inmediato, no se requiere de alguna descarga, configuración o instalación; el usuario puede mantenerse conectado a la aplicación web mediante algún navegador de su preferencia y lo más importante es que varios usuarios podrán

estar conectados al mismo tiempo sin que tenga alguna interferencia debido a que son menos propensas a tener fallas técnicas. [1]

Dentro del artículo científico publicado en la Quinta Conferencia Nacional de Computaciones en la Guayana, Venezuela con título “Metodología para el desarrollo de Aplicaciones Web utilizando Datos Enlazados” sus autores Tania R., Ricardo D., José A., indican que para la generación de una aplicación web su desarrollo se basa en dos fases muy importantes las cuales son:

Primera Fase: la creación de un servicio web, puesto que en mencionada fase se elegirá lenguaje de programación, librerías, métodos, operaciones y consultas que sean necesarias según los requerimientos del sistema o empresa en la que se vaya a realizar la implementación.

Segunda Fase: Primeramente, se deberá definir el protocolo que va a llegar la aplicación web para el consumo del servicio, posteriormente, teniendo los métodos claros y concisos, se procede a la creación de las distintas interfaces para que la navegación por el sitio sea amigable con el usuario. [2]

En el proyecto de tesis desarrollado por Edison Vega en la Universidad de las Fuerzas Armadas ESPE con el tema “Desarrollo de un sistema de Geolocalización para monitorear los vehículos de transporte de la cooperativa Andina” menciona que los Sistemas de Información Geográfica (GIS) son un instrumento que facilitan la organización y gestión de la información geográfica de objetos o personas en sus labores diarias, para que de esa manera colabore en la toma de decisiones.

En redes es muy utilizado debido a que se toma dos puntos (inicial y final) para obtener una ruta más óptima, teniendo en cuenta horarios, circulación de vehículos y direcciones, así como también se puede ir incorporando más utilidades como límites de velocidad en el trayecto, siendo estos de gran ayuda para obtener un mejor cálculo en los tiempos de recorrido entre el punto inicial y final. [3]

Leonardo Mancheno en su proyecto con tema “Aplicación web para la Gestión del Departamento Técnico de reparación y garantías de la empresa Telnet soporte en la ciudad de Ibarra” menciona que en la actualidad se han ido implementando soluciones informáticas mediante servicio web para la gestión de toda la información de empresas asociadas a la prestación de algún tipo de servicio, en dicho caso menciona un departamento de Soporte Técnico el cual no ha llevado durante toda su trayectoria algún sistema que beneficie la gestión del mismo. La inserción de nuevas tecnologías a las empresas para la gestión de la información mejora los tiempos de respuesta debido a que dicha información está mejor organizada y de fácil acceso para los usuarios. [4]

### **1.2.1 Contextualización del Problema**

En América Latina las Pequeñas y Medianas Empresas (PYMES) representan el 65% de la economía y existe una gran necesidad de que puedan iniciar con nuevas tecnologías en cada negocio para hallar oportunidades en nuevos mercados y poder conservarlos por más tiempo. La mayoría de las empresas están buscando soluciones tecnológicas que le ayuden con ciertos problemas que se han presentado hasta el momento y los que quisieran mejorar, como el incremento de ventas y productividad para así tener una reducción de costos dentro de los procesos, pero una cierta parte no han logrado encontrar una solución a su medida. Otra parte de estas empresas ni siquiera desean añadir recursos tecnológicos ya sea por la falta de conocimiento o el miedo a perder información importante del negocio.[5]

A partir de que inicio la pandemia del COVID 19 en Ecuador las PYMES empezaron a agregar nuevas tecnologías en sus negocios para poder mantenerse de pie o seguir avanzando. Todo esto para poder realizar trabajos desde casa, compras y ventas por internet, requerimiento de un servicio específico, capacitaciones, entre otras. Ciertos negocios empezaron a darle un mayor uso a las redes sociales y en pocos casos adoptaron la creación de un sitio web con funcionalidad de acuerdo con el mismo.[6]

Para las empresas que ofrecen servicios dentro de Ecuador no tienen muchas oportunidades debido a los altos costos de implementación de nuevas tecnologías, al



ser este problema uno de los principales dificultades que las empresas tengan más oportunidades de crecimiento, sin embargo el gobierno intenta apoyar a dichas empresas con oportunidades financieras, el hecho de tener un crédito es demasiado esfuerzo para propietarios y empleados en caso de tenerlos, muchos de los gerentes no deciden tener una deuda debido a que piensan que la situación no da para aquello pues la economía del país ha estado muy difícil en los últimos años.

Dentro de la provincia de Tungurahua la innovación es desarrollo en las empresas que se dedican a servicios, solo hasta el 2014 esta provincia contaba con 122 empresas de este tipo, porque en la actualidad ya deben existir aún más estamos hablando de que la provincia siempre ha tratado de innovar con empresas que den un mejor servicio y tener una propuesta de valor que a su empresa le haga crecer, para quienes desean empezar a crear un negocio innovador empresas públicas y privadas brindan un apoyo inicial al mejor proyecto. [7]

La empresa EPC Compu dedicada a las reparaciones de computadoras y accesorios, se encuentra ubicada en la ciudad de Ambato, teniendo clientes y proveedores potenciales ha logrado mantenerse en el mercado a pesar de la alta competitividad, pero debido a la mala gestión de soporte técnico en ciertas ocasiones ha causado pérdidas económicas. Teniendo en cuenta el problema antes mencionado se ha visto en la necesidad de implantar una solución eficaz al mismo, por tanto, se propone un Sistema Web para la Gestión del Servicio Técnico aplicando georreferenciación para la recepción y entrega de dispositivos tecnológicos.

En la actualidad se requiere un servicio a domicilio en la mayor parte de empresas dentro de la ciudad de Ambato, debido a que muchas de las personas salen a sus trabajos o deben estudiar y no tienen el tiempo necesario para acercarse a un local físico que dispongan de soporte técnico para dispositivos tecnológicos por ello se construirá un sistema web en el cual puedan coordinar la entrega y recepción de dispositivos a domicilio aplicando georreferenciación. Dichos clientes necesitan estar siempre informados del estado de sus dispositivos durante todo el proceso tales como Recepción, diagnóstico, reparaciones o mantenimiento y entrega el cual se lo podrá notificar por medio del sistema.

### **1.2.2 Delimitación**

**Área académica:** Software

**Línea de investigación:** Desarrollo de Software

**Delimitación Espacial:** Empresa EPC Compu.

#### **Delimitación Temporal:**

La presente investigación se la va a realizar durante el ciclo académico Abril 2022 – Agosto 2022.

### **1.2.3 Justificación**

La presente investigación tiene como propósito ayudar a la empresa EPC Compu con la implementación de un Sistema Web para la Gestión de Servicio Técnico como reparación, recepción y entrega de dispositivos tecnológicos aplicando georreferenciación en la Empresa EPC Compu.

El sistema le otorgara una mejor organización a EPC Compu para gestionar de mejor manera la atención a sus clientes con un mejor tiempo de respuesta en todos sus requerimientos, previniendo un déficit en el servicio ofrecido. La gestión de soporte técnico de dispositivos informáticos tendrá un aspecto positivo hacia la empresa debido a que no solamente lo manejará la misma sino también sus clientes para que EPC Compu siga creciendo con mayor flexibilidad gracias a las nuevas innovaciones tecnológicas.

En la actualidad EPC Compu atiende a sus clientes a través de dispositivos móviles y redes sociales mismas que carecen de una página informativa que le ayude al cliente a tener una idea clara de todos los servicios que ofrece. Incluso la gran mayoría de clientes desconocen el servicio de recolección y entrega a domicilio, por ello se va a

utilizar la georreferenciación dentro del sistema, para que de esa manera el cliente pueda elegir el horario más conveniente.

- **Factibilidad Técnica:**

Técnicamente el proyecto es factible debido a que se tiene los recursos tecnológicos y conocimientos necesarios y suficientes para la investigación.

- **Factibilidad Operativa:**

El proyecto es factible operativamente debido a que se contará con el apoyo de la Empresa EPC Compu..

- **Factibilidad Económica:**

El financiamiento del estudio será por parte del investigador.

## **1.2.4 Fundamentación Teórica**

### **1.2.4.1 Tecnología de la Información**

El inicio de la tecnología de la información con el fin de distinguir entre las máquinas con un alcance limitado de otras que tienen un propósito más general. Se basa en la investigación y el desarrollo de sistemas de información tales como aplicaciones de software y hardware. En pocas palabras, un departamento de Tecnologías de la Información se asegura que todas las computadoras funciones perfectamente en toda la estructura realizada.[8]

Cubre ciertas áreas como la administración, diseño y soporte de sistemas informáticos y telecomunicaciones, desarrolladores de software, informáticos y gestión de redes. [8]

#### **1.2.4.1.1 Tecnologías de la Información Geográfica (TIC)**

Es el conjunto de métodos y técnicas tanto clásicos como modernos referentes a la Cartografía como la leer, interpretar y elaborar mapas mediante las siguientes técnicas:

Fotointerpretación, en la cual se puede estudiar la evolución que tiene el territorio mediante fotos aéreas tomadas de manera vertical desde aviones.[9]

Teledetección, se debe estudiar e interpretar la información que emiten los satélites alrededor del planeta mediante claves proporcionadas por los propios satélites. [9]

Además de sistemas de Información Geográfica los cuales sirven para realizar un análisis espacial y de esa manera se pueda realizar o construir los mapas digitales que tenemos en la actualidad. [9]

#### **1.2.4.2 Sistema de Información Georreferencial**

Es aquel marco de trabajo en el cual se debe adjuntar, gestionar y analizar todos los datos obtenidos. Millones de aplicaciones y organizaciones dan utilidad a estos sistemas para establecer mapas de comunicación, que comparten y analizan información para resolver distintos problemas alrededor del mundo. Para la realización de estos sistemas se toman en cuenta cuatro variables o puntos importantes como son:

##### **Datos**

la mayoría de ellos provienen de un componente geográfico como imágenes, atributos y mapas. [10]

##### **Mapas**

Son el depósito geográfico para la capa de datos y dichos análisis con lo que se va a trabajar. [10]

##### **Análisis**

El análisis espacial le permite evaluar, estimar y predecir, interpretar y comprender, y más en función de la idoneidad y la capacidad, proporcionando nuevas perspectivas sobre su conocimiento y toma de decisiones. [10]

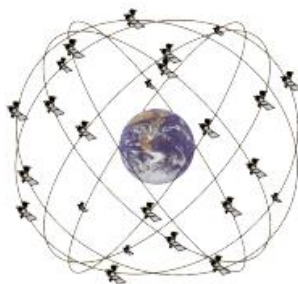
### **Aplicaciones**

La aplicación brinda una experiencia de usuario enfocada que hace que GIS sea accesible para todos. Funcionan en cualquier dispositivo: teléfono, tableta, navegador web y escritorio.[10]

#### **1.2.4.3 Sistema de Posicionamiento Global**

Se desarrollo en Estados Unidos por el Departamento de Defensa, en el programa NAVSTAR. Quienes lo empezaron a utilizar durante los primeros años fueron barcos, aviones y unidades terrestres, los cuales se dieron cuenta que fue de gran utilidad para realizar sus viajes. Este sistema está formado por una constelación de 5 estaciones en la superficie terrestre y 24 satélites. [11]

Dichas estaciones terrestres se encuentran en varios puntos de la tierra como en Hawái, Isla de Ascensión, Diego García, Atolón de Kwajalein y Colorado Springs, las cuales se encargan de toda la supervisión de la parte operativa de cada uno de los satélites. [11]



**Figura 1.** Representación sistema de posicionamiento global  
**Elaborado por:** Valeria Quispe

#### **1.2.4.4 Geolocalización**

Hace referencia a la ubicación geográfica de un objeto en específico que se lo puede observar mediante distintos dispositivos tecnológicos como teléfonos inteligentes, computadoras, ordenadores portables, entre otros. Todo esto se lo puede realizar mediante un GPS integrado en el objeto o dispositivo. [12]

##### **1.2.4.4.1 Api de Geolocalización**

En la actualidad la ubicación se está convirtiendo en el día a día de muchas aplicaciones para poderlos compartir y en muchos de los casos ver un recorrido. El más utilizado es la API de JavaScript de Google Maps debido a que en muchas páginas informativas tiene su ubicación fijada en un mapa para que quienes se encuentren interesados tengan un mayor espectro de referencia y llegar con mayor facilidad. Este api la podemos encontrar en código abierto y pagado como OpenLayers, Leaflet, Bing Maps, Map Quest y Here Maps, sin embargo, la Api de JavaScript de Google Maps tiene su límite gratuito con muchas funcionalidades que será suficiente para el desarrollo de aplicaciones web.[12]

##### **1.2.4.4.1.1 Precios de Api de Geolocalización según su funcionalidad**

###### **Mapas**

SKU	\$200 de crédito mensual Uso gratuito equivalente	Rango de volumen mensual (Precio por miles)		
		De 0 a 100,000	De 100,001 a 500,000	Más de 500,001
Mobile Native Static Maps	Cargas ilimitadas	\$0.00	\$0.00	
Mobile Native Dynamic Maps	Cargas ilimitadas	\$0.00	\$0.00	
Embed	Cargas ilimitadas	\$0.00	\$0.00	
Embed Advanced	Hasta 14,000 cargas	\$14.00	\$11.20	
Static Maps	Hasta 100,000 cargas	\$2.00	\$1.60	COMUNICARSE CON EL EQUIPO DE VENTAS para obtener descuentos por volumen
Dynamic Maps	Hasta 28,000 cargas	\$7.00	\$5.60	
Prueba Beta de Local Context Map	Es necesario habilitar Dynamic Maps	\$0.00 durante la prueba Beta	\$0.00 durante la prueba Beta	
Static Street View	Hasta 28,000 panorámicas	\$7.00	\$5.60	
Dynamic Street View	Hasta 14,000 panorámicas	\$14.00	\$11.20	

**Figura 2.** Precios de Api de Geolocalización en función de los mapas  
**Fuente:** Google [34]

SKU	\$200 de crédito mensual Uso gratuito equivalente	Rango de volumen mensual (Precio por miles)		
		De 0 a 100,000	De 100,001 a 500,000	Más de 500,001
Directions	Hasta 40,000 llamadas	\$5.00	\$4.00	
Directions Advanced	Hasta 20,000 llamadas	\$10.00	\$8.00	
Distance Matrix	Hasta 40,000 elementos	\$5.00	\$4.00	COMUNICARSE CON EL EQUIPO DE VENTAS para obtener descuentos por volumen
Distance Matrix Advanced	Hasta 20,000 elementos	\$10.00	\$8.00	
Roads - Route Traveled	Hasta 20,000 llamadas	\$10.00	\$8.00	
Roads - Nearest Road	Hasta 20,000 llamadas	\$10.00	\$8.00	
Roads - Speed Limits	Hasta 2,000 elementos	\$20.00	\$16.00	

**Figura 4.** Precios de Api de Geolocalización en función de las rutas  
**Fuente:** Google [34]

## Rutas

## Sitios

SKU	\$200 de crédito mensual Uso gratuito equivalente	Rango de volumen mensual (Precio por miles)		
		De 0 a 100,000	De 100,001 a 500,000	Más de 500,001
Solicitudes a Autocomplete				
Autocomplete, por solicitud	Hasta 70,000 solicitudes	\$2.83	\$2.27	COMUNICARSE CON EL EQUIPO DE VENTAS para obtener descuentos por volumen
Place Autocomplete, por solicitud. Por lo general, va seguida de una llamada de Places Details. Consulta los precios a continuación.				
Query Autocomplete, por solicitud	Hasta 70,000 solicitudes	\$2.83	\$2.27	

**Figura 6.** Precios de Api de Geolocalización en función de los sitios [33]  
**Fuente:**

#### **1.2.4.5 Sistema Web**

Es aquel en el cual los usuarios pueden tener acceso a través de un servidor web lo mismo que accede desde internet en un navegador. Son de mayor utilidad debido a que son fáciles de manejar, ahorran costos de hardware y software. En el ámbito colaborativo y trabajo a distancia les trae muchos beneficios puesto que no necesitan obligatoriamente llevar un dispositivo adicional para el software lo pueden acceder desde cualquier lugar en cualquier dispositivo. [13]

##### **1.2.4.5.1 Aplicaciones Web**

Son programas alojados en un servidor web para poder manejarlos a través de un navegado. Cada uno de los datos a los que se tiene acceso están almacenados en la web para ser procesados en cada interfaz. La arquitectura utilizada en este tipo de aplicaciones es Cliente-Servidor, de esa manera se pueda compartir sus scripts tanto del lado del servidor como en la gestión de la información para que el cliente la pueda observar de manera entendible. [13]

###### **1.2.4.5.1.1 Ventajas de las Aplicaciones Web**

Ahorro de tiempo: Debido a la sencillas de la gestión debido a que en la mayoría de los casos evitan realizar instalación de programas adicionales.

Actualización continua e inmediata: La versión de la aplicación únicamente se lo actualiza desde el desarrollador principal por ende los usuarios no tendrán que pasar demasiado tiempo esperando una actualización.

Recuperación de Datos: Los usuarios puedes tener acceso a todos los datos desde cualquier lugar y no necesitan llevar obligatoriamente un ordenador portátil, salvo casos excepcionales donde deba realizar una presentación. Pero incluso desde un dispositivo móvil pueden tener acceso. [14]



Compatibilidad: El hecho de que los navegadores siempre están actualizados, la aplicación siempre se la podrá manejar con sus versiones sin necesidad de tener un navegador en específico para el manejo de la aplicación.

Ahorro de Hardware: Las aplicaciones web tienden a no consumir tantos recursos el hecho de estar alojados en un servidor web y no necesita de hardware en un ordenador más que un simple navegador para tener acceso. [14]

#### **1.2.4.5.1.2 Aplicaciones Web con Diseño Adaptable**

Se denomina Diseño Adaptable al conjunto de técnicas tanto en el diseño como en la parte de desarrollo de una aplicación web, con la finalidad de que se pueda visualizar de la misma manera en todos los dispositivos sin necesidad de realizar cambios para distintos dispositivos. De esa manera el usuario tendrá una experiencia agradable al momento de utilizar la aplicación independientemente del dispositivo que vaya a utilizar. [15]

#### **1.2.4.5.1.3 Aplicaciones Web Móviles**

Se puede decir que son las aplicaciones con el menor coste de desarrollo, incluso el tiempo en que él se desarrolla es más corto que aplicaciones móviles. No necesariamente se la debe alojar en una tienda de aplicaciones como App Store o Play Store, el único requerimiento es tener un navegador y acceso a internet. [16]

#### **1.2.4.5.1.4 Arquitectura de Aplicaciones Web**

Las aplicaciones Web en su arquitectura se encuentra tres componentes muy importantes como el servidor web, conexión a internet y unos o más clientes. El servidor Web el componente en el cual se distribuye la información según el requerimiento realizado a través una conexión a internet por medio del protocolo HTTP para que enviara de vuelta la información solicitada. Estas se encuentran

basadas en el modelo Cliente/Servidor debido a la gestión que realizan los servidores web y que ocupan como interfaz páginas web. [17]

#### **1.2.4.6 Tecnologías de Desarrollo Web**

Las tecnologías para el desarrollo web cumplen la función de crear la interfaz que se mostrara al usuario y así de esa manera tenga una comunicación directa con el servidor y alojarlo. Incluso darle un formato adecuado según el tipo de sistema que se vaya a implementar.

##### **1.2.4.6.1 NET**

.NET es una plataforma de desarrollo de código abierto, multiplataforma y gratuita, la cual ha sido creada por Microsoft para la construcción de varios tipos de aplicaciones. [18]

Dentro de la plataforma se utilizan varios lenguajes de programación, implementaciones, herramientas y bibliotecas para la creación de aplicaciones. .NET tiene varias ventajas en los entornos de desarrollo como la independencia del lenguaje utilizado, biblioteca de clases, seguridad e interoperabilidad. [19]

##### **1.2.4.6.2 Laravel**

Es un framework para el desarrollo de aplicación utilizando el lenguaje de programación PHP, para quienes se adapten será un gran compañero durante todo el proceso de desarrollo puesto a que es de tipo MVC (Modelo-Vista-Controlador) para que de esa manera en cierto punto el usuario no se preocupe de ciertos temas como la instancia de clases y métodos que necesitan a lo largo del desarrollo sin la necesidad de tener que repetirlo demasiadas veces. [20]

El futuro de laravel es bastante prometedor debido a que ha ganado espacio dentro de profesionales quienes lo utilizan con mayor frecuencia en sus proyectos de desarrollo, en ciertas empresas con solo tener un año de experiencia en este framework te asegura un puesto y sobre todo tener el acceso de crear nuevos proyectos increíbles. [20]

#### **1.2.4.7 Sistemas Gestores de Base de Datos**

Sistemas en los cuales se puede crear, gestionar y administrar una base de datos para su respectivo almacenamiento y realizar la búsqueda de información de manera eficiente. Existen varios SGBD que se lo ocupa de acuerdo con la información y el uso que se le vaya a dar, los cuales son:

- **MySql**

Hace referencia a una base de datos con licencia publica más conocida por sus siglas como GNU, implementado por un modelo cliente/servidor en cual lo puedes instalar sin mayor inconveniente y adaptarlo a las necesidades del requerimiento. No se recomienda utilizar esta base de datos cuando se tiene grandes volúmenes de información. [21]

- **PostgresSQL**

Uno de los gestores de base de datos relacionales que está orientado a objetos y soporta gran variedad de tipo de datos, es de libre acceso y código abierto la cual disponen de un gran equipo de desarrolladores los cuales ayudan a solventar problemas y realizar mejoras en el software. [21]

##### **1.2.4.7.1 SQL Server**

Es un sistema gestor de base de datos más utilizado en el mercado debido a que su servicio se lo puede consumir desde una gran variedad de aplicaciones de software con gran cantidad de datos para administrarlos sin mayores inconvenientes, ya que su

interfaz es fácil de manejar. Este sistema dispone de grados funcionalidad como las herramientas que son destinadas a la gestión y análisis de toda la información especialmente se ha visto que lo ocupan en la inteligencia empresarial para tener datos mediante machine learning y así conocer de mejor manera el negocio y clientes. [22]

#### **1.2.4.8 Metodologías de Desarrollo de Software**

Es de gran ayuda para el diseño de una solución de software que lo realiza mediante un conjunto de técnicas y métodos, se lo realiza para tener una mejor organización al momento de ir realizando el proyecto de esta manera tener un mayor control del trabajo y cumplir con todos los requerimientos planteados de manera disciplinada. [23]

##### **1.2.4.8.1 Ventajas del Uso de una Metodología de Desarrollo de Software**

Existen grandes beneficios de utilizar una metodología al momento de realizar desarrollo de software uno de los principales objetivos es ayudar al equipo de trabajo para que tengan un mayor enfoque de las tareas que se deben realizar y el cliente obtenga beneficios para entregar un producto de calidad.

Entre las ventajas más importantes que se puede mencionar son:

Productos de mayor calidad, debido al momento en que se aplica las metodologías se realizan pruebas durante el ciclo de desarrollo para que se puedan aplicar correcciones de errores de manera rápida y concreta. [24]

Clientes satisfechos, durante el proceso de desarrollo se lo mantiene informado al cliente de todos los avances realizado y pueda verificar si van acorde a sus requerimientos, de esa manera todo marche de manera flexible y clara con el cliente. [24]

Reducción de riesgos, a medida que se va realizando pruebas y acompañamientos con los clientes tiende a eliminar toda posibilidad de que al final el producto falle de manera absoluta. [24]

#### **1.2.4.8.2 Clasificación de Metodologías**

##### **1.2.4.8.2.1 Metodologías Tradicionales**

Este tipo de metodologías definen de manera completa y rigurosa todos los requerimientos desde un inicio, por ende, dentro de los procesos no se pueden realizar cambios de manera flexible y manejable. Las etapas están organizadas de manera que, si no se termina una, no puedes seguir con la siguiente etapa.

Las principales metodologías tradicionales que se puede mencionar son:

#### **Waterfall o Cascada**

Cada etapa dentro de esta metodología se la realiza de arriba hacia abajo obedeciendo todos los requerimientos solicitados, se realiza las respectivas pruebas y una vez que esté funcionando en perfectas condiciones se debe pasar a la siguiente etapa.

#### **Prototipado**

Se realiza un prototipo de manera rápida para que el usuario tenga una idea de cómo va a funcionar su software, para que así el cliente pueda decir lo que está bien o lo que está mal para que el producto surja de la mejor manera y sobre todo brindar un producto de calidad que le agrade al cliente.

#### **Espiral**

Es una mezcla entre dos metodologías cascada y prototipado, la diferencia es que dentro de sus etapas existe el análisis de riesgo para que sea de manera se vaya

procesando todas las etapas en forma de espiral. Dicha metodología se divide en cuatro etapas que son: planificación, análisis de riesgo, desarrollo de prototipado y evaluación del cliente.

#### **1.2.4.8.2.2 Metodologías Agiles**

Son las metodologías más utilizadas en los últimos años debido a la gran flexibilidad y agilidad para la realización de todo un gran proyecto, de paso ayuda a un equipo de trabajo a ser más organizados y productivos para poder adaptar el producto a los nuevos requerimientos que vayan surgiendo en el transcurso de la elaboración.

Existen gran variedad de metodologías giles sin embargo se nombrará las más importantes que son:

##### **Kanban**

Dicha metodología fue creada por la empresa automovilística Toyota. En la cual se va organizando todas las tareas en un tablero para ir las organizando en tareas pendiente, realizada y en proceso. De tal manera que las tareas se pueden ir priorizando según la necesidad incluso para tener un mayor orden y ver el avance del proyecto.

##### **Scrum**

Existen cuatro etapas las cuales se deben ir cumpliendo de acuerdo con los requerimientos, empieza por la planificación de la iteración o planning sprint, ejecución o sprint, reunión diaria o daily meeting y demostraciones de resultados o sprint view. Las etapas se realizan en bloques con un tiempo de entre dos a cuatro semanas para poder conseguir el resultado esperado en cada tarea asignada.

##### **Programación Extrema o XP**

Se basa en las relaciones interpersonales para el equipo de trabajo incluso con el cliente, su objetivo principal es mantener un buen ambiente laboral entre todos cumpliendo así doce aspectos importantes que se debe tener en cuenta los cuales son:

- Diseño sencillo
- Testing
- Refactorización y codificación con estándares
- Propiedad colectiva del código
- Programación en parejas
- Integración continua
- Entregas semanales e integridad con el cliente
- Cliente in situ
- Entregas frecuentes
- Planificación [25]

#### **1.2.4.9 Control de Calidad**

El control de calidad es un proceso que ayuda a una empresa a operar con éxito. Por ejemplo, si se detecta un problema, se realizan diversos estudios para verificar si el producto o servicio se realiza con la calidad correspondiente. El proceso completado a través de estándares de certificación, estos estándares son procedimientos muy importantes para cualquier empresa u organización que busque posicionarse en el mercado.

#### **1.2.4.10 Optimización de Recursos**

Cuando se refiere a la optimización de recursos, quiere decir al uso de toda la tecnología, producción y medios humanos de los que se apoya el proceso productivo para lograr el máximo beneficio. Esta optimización de recursos se consigue aportando cada uno de ellos, lo que le corresponde, sin tener que llevarlos al límite. Por lo tanto, una empresa será productiva cuando obtenga los mejores resultados, no porque tenga la mayor cantidad de recursos.

#### **1.2.4.11 Gestión del Proceso de Servicio Técnico**

Empresas que ofrecen servicio técnico, deben enfrentar varios retos, uno de ellos las llamadas continuas y desesperadas de clientes que requieren un diagnóstico rápido y oportuno, en el cual toda la empresa debe organizarse y optimizar los tiempos de salida de cada dispositivo tecnológico. La solución óptima para llevar a cabo este tipo de servicio es la creación de un software que les ayude a manejar mejor los clientes sobre ellos la venta de dicha aplicación es la eliminación de papel puesto que todo se lo digitalizara, se podrá gestionar de manera óptima cada una de las tareas y sobre todo mantener siempre informado al cliente sobre el estado de su requerimiento. Por último, la calidad de atención al cliente va a mejorar de manera eficiente.

### **1.3 Objetivos**

#### **1.3.1 Objetivo General**

Desarrollar un Sistema Web que ayude a una mejor organización dentro de la Gestión de Servicio Técnico aplicando georreferenciación para la recepción y entrega de dispositivos tecnológicos en la empresa EPC Compu.

#### **1.3.2 Objetivos Específicos**

- Definir el proceso que conlleva la parte de servicio técnico iniciando desde la recepción hasta la entrega del dispositivo en óptimas condiciones.
- Determinar la tecnología de desarrollo idónea para el proyecto.
- Definir la metodología de desarrollo ágil a utilizar en la implementación del sistema web.



- Implantar un Sistema Web para la Gestión de Servicio Técnico aplicando georreferenciación para la recepción y entrega de dispositivos tecnológicos en la empresa EPC Compu.

## **2. CAPÍTULO II.- METODOLOGÍA**

### **2.1 Materiales**

Durante el proceso de desarrollo del proyecto se realizó la investigación mediante el uso de fuentes bibliográficas verificadas en los cuales se incluyen tesis, libros, revistas, artículos científicos entre otros para conseguir toda la información necesaria que se refiere al desarrollo de un sistema web para la gestión de servicio técnico, aplicando Georreferenciación para la recepción y entrega de dispositivos tecnológicos, en el cual también se aplicó cada uno de los conocimientos aprendidos durante el proceso de estudio para el manejo de metodologías ágiles durante el proceso de desarrollo web además de la utilización de georreferenciación.

#### **Encuesta para clientes**

##### **Modelo de la Encuesta**

#### **¿Con que frecuencia utiliza un dispositivo tecnológico?**

##### **Elegir una sola opción**

- Todos los días
- Dos veces por semana
- Cuatro veces al mes
- Ningún día

#### **¿Qué tipo(s) de dispositivo(s) tecnológico(s) ha utilizado en su vida cotidiana?**

##### **Elegir varias opciones**

- Computadora
- Laptop
- Smartphone
- Tablet

**¿Cuál es el tiempo estimado de respuesta ante algún incidente por parte de Soporte Técnico?**

**Elegir una sola opción**

- Menor a 24 horas
- Mayor o igual a 24 horas

**¿Considera usted eficiente el proceso actual de recepción y entrega de los dispositivos tecnológicos?**

**Elegir una sola opción**

- Sí
- No

**Actualmente, para obtener información referente al soporte técnico lo realiza mediante:**

**Elegir una sola opción**

- Llamada Telefónica
- Visita Personal
- Mensaje de Texto
- Otro:

**¿Cree que una herramienta tecnológica, permita mejorar el proceso de recepción y entrega de los dispositivos tecnológicos?**

**Elegir una sola opción**

- Sí
- No

**¿Qué dispositivo(s) informático(s) le gustaría utilizar para obtener información referente al proceso de soporte técnico?**

**Elegir varias opciones**

- Computador
- Laptop
- Smartphone
- Tablet

**¿Le gustaría obtener información sobre el estado actual de su dispositivo tecnológico y el proceso a realizar?**

**Elegir una sola opción**

- Sí
- No

### **Modelo de la Entrevista al Personal**

#### **Parte I: Identificando perfil de usuario/cliente**

*Rol en organización:*

*Responsabilidades del entrevistado:*

#### **Parte II: Evaluación del problema**

*¿Qué problemas debe solucionar el Sistema?*

*Cómo se resuelven ahora*

*Cómo los resolvería*

#### **Parte III: Entendiendo el Entorno**

*Tienen los usuarios experiencia en este tipo de aplicaciones*

*Hay aplicaciones/sistemas relevantes para tener en cuenta en nuestro Sistema*

*Cuáles son las expectativas de usabilidad del Producto*

*¿Cuáles son las expectativas de formación a usuarios para el uso del Sistema?*

*¿Qué tipo de ayuda requerirá el usuario?*

#### **Parte IV: Evaluando la oportunidad**

*¿Quién en la organización necesita la aplicación?*

*¿Cuántos tipos de usuarios usarán la aplicación?*

*¿Cómo valoraría que la solución ha sido un éxito?*

#### **Parte V: Evaluando la solución**

Resumen de las capacidades clave de la solución propuesta:

*¿Añadiría alguna?*

*¿En qué posición de un ranking pondría cada una de estas capacidades?*

#### **Parte VI: Evaluando las necesidades de fiabilidad, rendimiento y soporte**

*¿Cuáles son tus expectativas en cuanto a fiabilidad del sistema?*

*¿Cuáles son tus expectativas en cuanto a rendimiento del sistema?*

*¿Dará soporte el ayuntamiento al producto? ¿O lo hará otro equipo?*

*¿Existen necesidades especiales para dar el soporte?*

*¿Cuáles son los requisitos de instalación y configuración?*

*¿Cómo se distribuirá el software?*

## **Parte VII: Recapitulación**

Después de la entrevista y mientras los datos están frescos haz un resumen de las principales necesidades y problemas identificados por el usuario:

## **2.2 Métodos**

### **2.2.1 Modalidad de investigación**

#### **Modalidad bibliográfica**

Se utilizó libros, documentos, artículos, revistas, entre otros para la realización del marco teórico acerca de la implementación un Sistema Web para la Gestión de Servicio Técnico como reparación, recepción y entrega de dispositivos tecnológicos aplicando georreferenciación en la Empresa EPC Compu.

#### **Modalidad de campo**

Debido a que se realizó un análisis de los procesos que conlleva la gestión de Servicio técnico iniciando desde la recepción del dispositivo hasta la entrega de este en óptimas condiciones luego de haber sido reparado.

### **Modalidad aplicada**

Es considerada dicha modalidad como aplicada, puesto que se ocupó cada uno de los conocimientos adquiridos en el estudio de la carrera para resolver el problema presentado.

### **2.2.2 Población y muestra**

La población que se consideró para la investigación fue todo el personal que labora actualmente en EPC COMPU como Gerente, secretaria, soporte técnico y contabilidad, también se incluye a sus clientes actuales dando un total de 205 como se muestra en el detalle de la Tabla1.

**Tabla 1.** Definición de la población de Estudio  
**Elaborado por:** Valeria Quispe

<b>N.º</b>	<b>Población</b>	<b>Numero</b>	<b>Porcentaje</b>
1	Gerente	1	0.5%
2	Secretaria	2	0.5%
3	Soporte Técnico	2	1%
4	Contabilidad	1	0.5%
5	Clientes	200	97.5%
<b>Total</b>		<b>205</b>	<b>100%</b>

Se realizó el cálculo de la muestra debido a que el número de población es 205 y sobrepasa los 100, por ende, se realizara con un nivel de confianza del 95% dando como resultado una muestra de 134, como indica en la Tabla 2 con todas las variables detalladas.

**Tabla 2.** Definición de la muestra a Evaluar  
**Elaborado por:** Valeria Quispe

<b>Variable</b>	<b>Valor</b>
E	0.05
N	205
Ö	0.5
Confianza	95
Área a la izquierda de -Z	0.025
Z	-1.96
-Z	1.96
<b>Muestras (n)</b>	<b>134</b>

### 2.2.3 Recolección de la información

La técnica que se utilizó para la recolección de la información fue a través de encuestas y entrevistas al personal de la empresa, para lo cual se utilizó como instrumento el cuestionario de preguntas cerradas.

**Tabla 3.** Recolección de la información  
**Elaborado por:** Valeria Quispe

<b>Preguntas Básicas</b>	<b>Explicación</b>
¿Para qué?	Para alcanzar los objetivos de la investigación
¿De qué personas u objetos?	Gerencia  Secretaría  Soporte técnico

	Contabilidad Clientes
¿Sobre qué aspectos?	Gestión de Soporte Técnico Mejora en tiempos de respuesta y recolecciones
¿Quién, Quiénes?	Investigador: Valeria Alexandra Quispe Pérez
¿Cuándo?	Abril – Septiembre 2022
¿Dónde?	EPC Compu
¿Cuántas veces?	Una
¿Qué técnicas de recolección?	Encuestas Entrevista
¿Con qué?	Cuestionario Cuestionario de preguntas abiertas

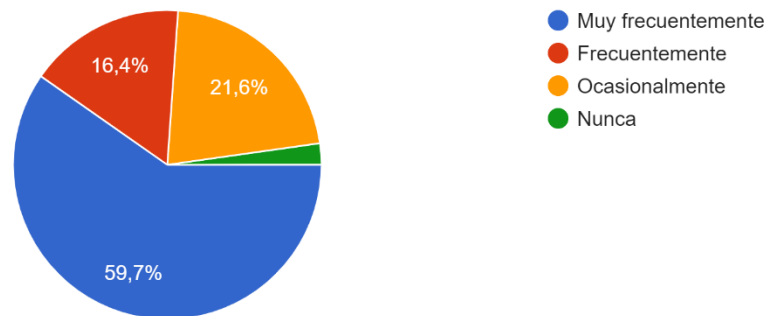
### Resultado de Encuesta Aplicada

#### 1. ¿Con que frecuencia utiliza un dispositivo tecnológico?

**Tabla 4.** Frecuencia con la que se utiliza un dispositivo tecnológico  
**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Muy frecuentemente	80	59.7%
Frecuentemente	22	16.4%
Ocasionalmente	29	21.6%
Nunca	3	2.3%





**Figura 7.** Frecuencia con la que se utiliza un dispositivo tecnológico  
**Elaborado por:** Valeria Quispe  
**Fuente:** Encuesta Aplicada

**Análisis e interpretación:** con respecto a los resultados de las encuestas aplicadas se evidencia que en su gran mayoría las personas utilizan de manera muy frecuente dispositivos tecnológicos es decir un total de 80 personas de las 134 que han sido encuestadas, mientras que el 16.4% afirma que utiliza frecuentemente, el 21.6% ocasionalmente y el 2.3% Nunca.

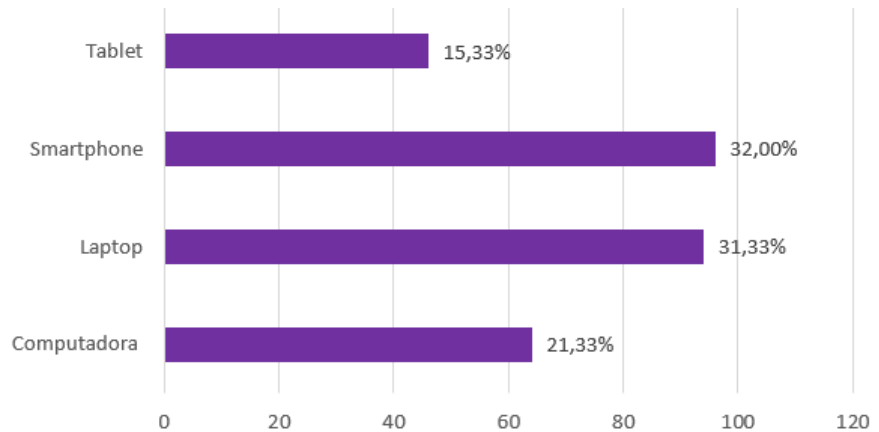
Se concluye que la gran parte de encuestados disponen de dispositivos tecnológicos, los cuales en algún momento necesitaran soporte técnico.

**2. ¿Qué tipo(s) de dispositivo(s) tecnológico(s) ha utilizado en su vida cotidiana?**

**Tabla 5.** Tipos de dispositivos tecnológico utilizados en la vida cotidiana

**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Computadora	64	21,33%
Laptop	94	31,33%
Smartphone	96	32,00%
Tablet	46	15,33%



**Figura 8.** Tipos de dispositivos tecnológico utilizados en la vida cotidiana

**Elaborado por:** Valeria Quispe

**Fuente:** Encuesta Aplicada

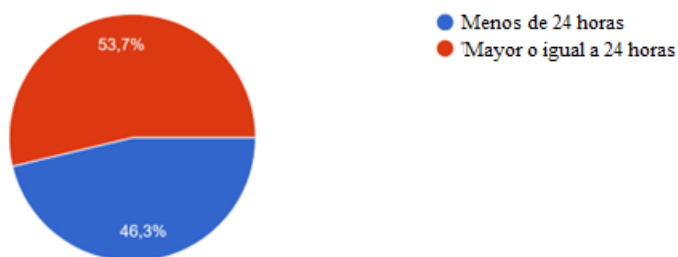
**Análisis e interpretación:** se puede evidenciar que en su gran mayoría utilizan dispositivos tecnológicos como Laptop y Smartphone con un 31.33% y 32% respectivamente, mientras que el 21.33% utiliza computadora de escritorio y el 15.33% utiliza una Tablet

Gran parte de los encuestados utilizan principalmente computadoras y laptops que es lo que se maneja en el soporte técnico de la empresa EPC Compu por ende se tiene una gran ventaja en dichos dispositivos.

3. ¿Cuál es el tiempo estimado de respuesta ante algún incidente por parte de Soporte Técnico?

**Tabla 6.** Atención por parte de soporte técnico en cuanto a incidentes  
**Elaborado por:** Valeria Quispe

Variables	Frecuencia	Porcentaje
Menos de 24 horas	62	46.3%
Mayor o igual a 24 horas	72	53.7%



**Figura 9.** Atención por parte de soporte técnico en cuanto a incidentes  
**Elaborado por:** Valeria Quispe  
**Fuente:** Encuesta Aplicada

**Análisis e interpretación:** Mediante la encuesta se puede evidenciar que los clientes en su gran mayoría opinan que el servicio que brinda EPC Compu se demora más de 24 horas lo que compete un 53.7%, por lo cual se dará una mejor solución al principal problema que se está evidenciando; mientras que el 46.3% menciona que el servicio ha sido atendido en menos de 24 horas.

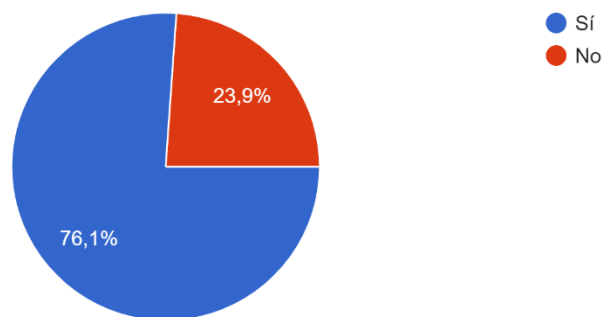
De acuerdo con los datos recabados mediante la encuesta opinan que el servicio es tarda más de lo esperado, por lo que buscan tener una mejor organización de todo el proceso que conlleva la empresa para manejar el servicio de soporte técnico.

**4. ¿Considera usted eficiente el proceso actual de recepción y entrega de los dispositivos tecnológicos?**

**Tabla 7.** Eficiencia en el proceso de recepción y entrega de los dispositivos tecnológicos

**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	102	76.1%
No	32	23.9%



**Figura 10.** Eficiencia en el proceso de recepción y entrega de los dispositivos tecnológicos

**Elaborado por:** Valeria Quispe

**Fuente:** Encuesta Aplicada

**Análisis e interpretación:** Se puede verificar que el servicio que entrega la empresa en este caso es eficiente, debido a que un total de 102 personas confirman dicha pregunta dando un total del 76.1% mientras que el 23.9% opina que no.

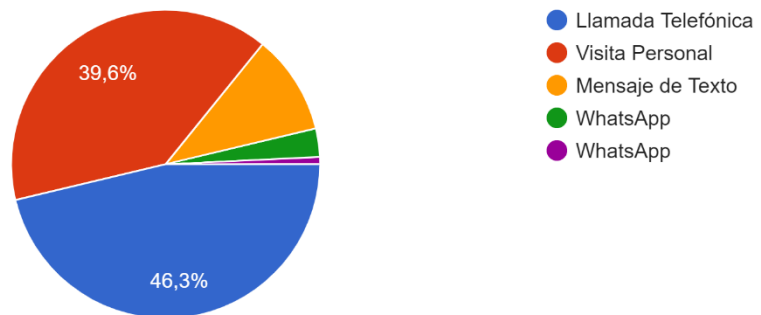
A pesar de que en la pregunta numero 3 opinan que el servicio es un poco lento, se puede verificar que el servicio es muy eficiente teniendo una razón más por la que la empresa debe seguir mejorando.

**5. Actualmente, para obtener información referente al soporte técnico lo realiza mediante:**

**Tabla 8.** Medios por los cuales soporte técnico obtienen información

**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Llamada Telefónica	62	46.3%
Visita Personal	53	39.6%
Mensaje de Texto	14	10.4%
Otros	4	3.7%



**Figura 11.** Medios por los cuales soporte técnico obtienen información

**Elaborado por:** Valeria Quispe

**Fuente:** Encuesta Aplicada

**Análisis e interpretación:** la gran mayoría de encuestados pueden obtener información acerca del soporte técnico de sus equipos mediante llamada telefónica con un 46.3%, mientras que visita personal con 36.9%, mensajes de texto con el 10.4% y otros como WhatsApp con el 3.7%.

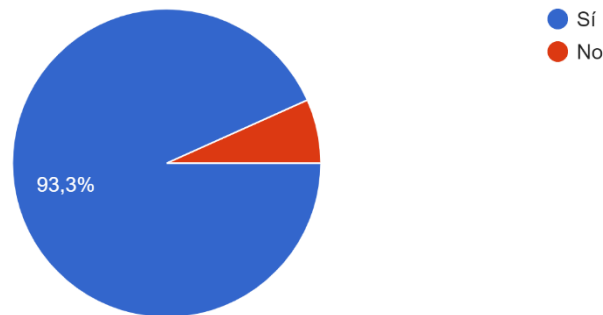
De acuerdo con los datos evidenciados se puede obtener una amplia información acerca de los distintos medios de por los cuales se obtienen los datos para poder brindar un eficiente soporte técnico.

**6. ¿Cree que una herramienta tecnológica, permita mejorar el proceso de recepción y entrega de los dispositivos tecnológicos?**

**Tabla 9.** Aceptación de herramientas tecnológicas para la mejora de procesos de entrega-recepción de Dispositivos

**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	125	93.3%
No	9	6.7%



**Figura 12.** Aceptación de herramientas tecnológicas para la mejora de procesos de entrega-recepción de dispositivos.

**Elaborado por:** Valeria Quispe

**Fuente:** Encuesta Aplicada

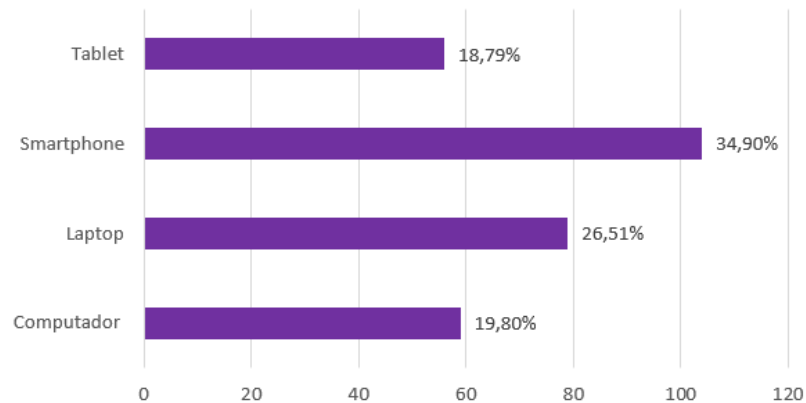
**Análisis e interpretación:** El 93.3% de los encuestados cree que una herramienta tecnológica va a ayudar a mejorar los procesos de la empresa mientras que el 6.7% cree que no.

El sistema que se está planteando dentro del presente proyecto será la mejor opción para este caso, debido a que están de acuerdo con la implementación de este.

**7. ¿Qué dispositivo(s) informático(s) le gustaría utilizar para obtener información referente al proceso de soporte técnico?**

**Tabla 10.** Lista de dispositivos informáticos para obtener información en el proceso de soporte técnico  
**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Computador	59	19,80%
Laptop	79	26,51%
Smartphone	104	34,90%
Tablet	56	18,79%



**Figura 13.** Lista de dispositivos informáticos para obtener información en el proceso de soporte técnico  
**Elaborado por:** Valeria Quispe  
**Fuente:** Encuesta Aplicada

**Análisis e Interpretación:** En si gran mayoría los encuestados prefieren recibir información mediante smartphone con un 34.90%, laptop con el 26.51% y computador con el 19.80%, mientras que con minoría opinaron por Tablet con un 18.79%

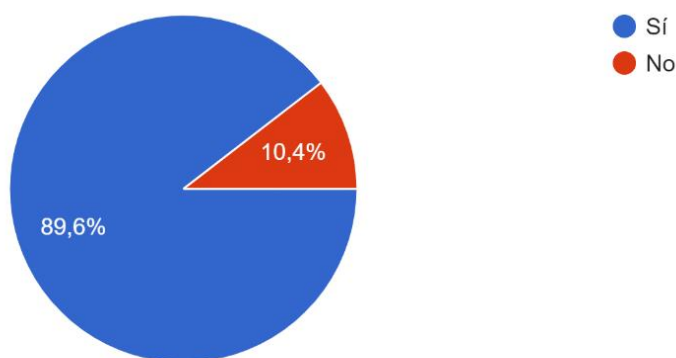
Se deberá implementar un sistema que ayude con dichos requerimientos para que funcione en smartphone, laptop y computadoras que son las principales opciones de acuerdo con esto se presenta un sistema web.

**8. ¿Le gustaría obtener información sobre el estado actual de su dispositivo tecnológico y el proceso a realizar?**

**Tabla 11.** Afirmación sobre la información sobre el estado de reparación del dispositivo tecnológico

**Elaborado por:** Valeria Quispe

<b>Variables</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Si	120	89.6%
No	14	10.4%



**Figura 14.** Afirmación sobre la información sobre el estado de reparación del dispositivo tecnológico

**Elaborado por:** Valeria Quispe

**Fuente:** Encuesta Aplicada

**Análisis e Interpretación:** Se puede evidenciar en la encuesta realizada que un total de 120 encuestados les gustaría obtener información mediante un dispositivo dando un total del 89.6%, mientras que el 10.4% dice que no.

Toda la información referente a la reparación de dispositivos se lo podrá visualizar a través del sistema a implementar para que tengan en cuenta todo el proceso y siempre se mantengan informados.



## **Resultado de la Entrevista Aplicada**

### **Objetivo de la Entrevista**

La presente entrevista tiene el objetivo de recolectar todos los requerimientos y aplicarlos en el sistema para que de esa manera se pueda tener un panorama más claro y conciso de lo que se desea hacer.

### **Entrevista a Secretaría**

#### **Parte I: Identificando perfil de usuario/cliente**

*Rol en organización:* Secretaría

*Responsabilidades del entrevistado:* Retiros, Verificar estado de los equipos, Brindar información al cliente, realizar seguimiento de los equipos, comunicación directa con los clientes.

#### **Parte II: Evaluación del problema**

*¿Qué problemas debe solucionar el Sistema?*

1. Ingresar información de los dispositivos
2. Dar la mayor parte de información por el sistema
3. Verificar todo el proceso al cual ingresa el dispositivo
4. Revisar el estado de retiros a domicilio

*Cómo se resuelven ahora*

1. La información de los dispositivos lo tienen en un Excel
2. Se llama al cliente para explicarle todo el proceso al que ingresa el dispositivo
3. Siempre se pregunta a los técnicos el estado de los dispositivos
4. Los motorizados envían un mensaje cuando es retirado el dispositivo.

### ***Cómo los resolvería***

1. Toda la información estará en el sistema
2. El cliente puede revisar todo el proceso en el sistema
3. Los técnicos subirán su respectivo informe al sistema
4. Se verifica los retiros que han realizado los motorizados

### **Parte III: Entendiendo el Entorno**

*Tienen los usuarios experiencia en este tipo de aplicaciones*

No tiene experiencia en este tipo de aplicaciones

*Hay aplicaciones/sistemas relevantes para tener en cuenta en nuestro Sistema*

No tienen más sistemas dentro de la empresa

*Cuáles son las expectativas de usabilidad del Producto*

Fácil manejo del sistema para evitar pérdidas de tiempo y brindar una mejor atención al cliente

*¿Cuáles son las expectativas de formación a usuarios para el uso del Sistema?*

La capacitación se debe realizar de manera estratégica debido a que unos los entenderán más rápido que otros

*¿Qué tipo de ayuda requerirá el usuario?*

El ingreso de toda la información y la visualización de reportes.

### **Parte IV: Evaluando la oportunidad**

*¿Quién en la organización necesita la aplicación?*

Secretaría, Soporte Técnico y Gerente

*¿Cuántos tipos de usuarios usarán la aplicación?*

5

*¿Cómo valoraría que la solución ha sido un éxito?*

El personal maneja de manera eficiente todo el sistema

### **Parte V: Evaluando la solución**

Resumen de las capacidades clave de la solución propuesta:

1. Ingresar el estado en el que se encuentran los dispositivos

2. Revisión de informe del diagnóstico realizado
3. Asignación de retiros
4. Elegir los productos que se van a realizar el diagnóstico
5. Mayor control a los motorizados
6. Organización de la información recabada por secretaría

*¿Añadiría alguna?*

Mapa de calor para verificar las zonas donde mayor acogida tiene la empresa para futuras opciones de sucursales.

*¿En qué posición de un ranking pondría cada una de estas capacidades?*

Se podría mencionar en el siguiente orden:

1. Organización de la información recabada por secretaría
2. Ingresar el estado en el que se encuentran los dispositivos
3. Revisión de informe del diagnóstico realizado
4. Mayor control a los motorizados
5. Elegir los productos que se van a realizar el diagnóstico
6. Asignación de retiros

## **Parte VI: Evaluando las necesidades de fiabilidad, rendimiento y soporte**

*¿Cuáles son tus expectativas en cuanto a fiabilidad del sistema?*

Tendrá la capacidad necesaria para solventar todas las necesidades de la empresa.

*¿Cuáles son tus expectativas en cuanto a rendimiento del sistema?*

Va a rendir de buena manera debido a que se tendrá toda la capacitación del sistema y dará muy buenos resultados para mejorar la organización de la empresa.

*¿Dará soporte el ayuntamiento al producto? ¿O lo hará otro equipo?*

Se realizará una reunión con todo el personal para verificar el tema.

*¿Existen necesidades especiales para dar el soporte?*

No

*¿Cuáles son los requisitos de instalación y configuración?*

Se utilizará cualquier navegador web

*¿Cómo se distribuirá el software?*

Es un sistema web en el cual tendrán acceso desde cualquier lugar y únicamente necesitan sus credenciales de ingreso.

## **Parte VII: Recapitulación**

Después de la entrevista y mientras los datos están frescos haz un resumen de las principales necesidades y problemas identificados por el usuario:

Necesidades para el sistema

1. Ingreso de información de clientes
2. Manipulación de la información
3. Gestión de retiros y entregas
4. Visualización de reportes
5. Mapa de calor de clientes

Funciones principales que se realiza en la Secretaría por lo que obligatoriamente debe estar dentro del producto final para que puedan llevar de mejor manera todo el proceso.

## **Entrevista a Soporte Técnico**

### **Parte I: Identificando perfil de usuario/cliente**

*Rol en organización:* Soporte Técnico

*Responsabilidades del entrevistado:* Realizar el diagnóstico, mantenimiento y reparaciones de dispositivos tecnológicos.

### **Parte II: Evaluación del problema**

*¿Qué problemas debe solucionar el Sistema?*

1. Falta de organización en la entrega de los dispositivos a los clientes finales.
2. El cliente final no tiene el precio aproximado una vez realizado el diagnóstico.
3. Comunicación lenta por parte de secretaría hacia los clientes.
4. Desconocen los días y horarios de recolecciones de dispositivos a domicilio.
5. Información organizada de manera convencional.

*Cómo se resuelven ahora*

1. Se realiza entregas de acuerdo con el horario que mencione el cliente.
2. Se le hace un cálculo una vez terminado la reparación o mantenimiento.

3. Únicamente se comunican por llamada telefónica o mensajería instantánea.
4. Se busca en un archivo de Excel o en un cuaderno de tareas.

### ***Cómo los resolvería***

1. Se establece los horarios de salida desde el sistema para que el cliente este informado.
2. Reporte de un valor aproximado de la reparación o mantenimiento.
3. Todo el proceso lo puede visualizar desde el sistema.
4. Las consultas se lo verificaran en el sistema según el requerimiento que necesite.

### **Parte III: Entendiendo el Entorno**

*Tienen los usuarios experiencia en este tipo de aplicaciones*

No tienen experiencia en este tipo de aplicaciones

*Hay aplicaciones/sistemas relevantes para tener en cuenta en nuestro Sistema*

No tienen más sistemas dentro de la empresa

*Cuáles son las expectativas de usabilidad del Producto*

Sea un sistema eficiente y sobre todo que sea fácil de manejar por parte de los distintos tipos de usuarios para que de esa manera la empresa tenga esa satisfacción de que el sistema funciona correctamente.

*¿Cuáles son las expectativas de formación a usuarios para el uso del Sistema?*

Mediante una buena capacitación organizada, será más factible el entendimiento del mismo y se lo debe realizar de manera independiente por departamento.

*¿Qué tipo de ayuda requerirá el usuario?*

Hasta manejarle bien el sistema, el llenado de los campos de datos y las consultas que se puede realizar.

### **Parte IV: Evaluando la oportunidad**

*¿Quién en la organización necesita la aplicación?*

Secretaría, Soporte Técnico y Gerente

*¿Cuántos tipos de usuarios usarán la aplicación?*

*¿Cómo valoraría que la solución ha sido un éxito?*

Una vez que se realicen las pruebas respectivas en la empresa en tiempo real

#### **Parte V: Evaluando la solución**

Resumen de las capacidades clave de la solución propuesta:

1. Ingresar el estado en el que se encuentran los dispositivos
2. Revisión de informe del diagnóstico realizado
3. Asignación de retiros
4. Elegir los productos que se van a realizar el diagnóstico
5. Mayor control a los motorizados
6. Organización de la información recabada por secretaría

*¿Añadiría alguna?*

Por el momento no

*¿En qué posición de un ranking pondría cada una de estas capacidades?*

Todas son importantes dentro del sistema

#### **Parte VI: Evaluando las necesidades de fiabilidad, rendimiento y soporte**

*¿Cuáles son tus expectativas en cuanto a fiabilidad del sistema?*

Se confía en que el sistema tendrá una excelente calidad y un fácil manejo.

*¿Cuáles son tus expectativas en cuanto a rendimiento del sistema?*

Se desea tener excelentes resultados y debido a que no son demasiados usuarios en línea se mantendrá siempre el sistema activo. Y se debe cumplir con todos los problemas antes mencionados.

*¿Dará soporte el ayuntamiento al producto? ¿O lo hará otro equipo?*

Eso lo decide el gerente de la empresa

*¿Existen necesidades especiales para dar el soporte?*

No

*¿Cuáles son los requisitos de instalación y configuración?*

Ingresar desde cualquier navegador web

*¿Cómo se distribuirá el software?*

Es un sistema web en el cual tendrán acceso desde cualquier lugar y únicamente necesitan sus credenciales de ingreso.

### **Parte VII: Recapitulación**

Después de la entrevista y mientras los datos están frescos haz un resumen de las principales necesidades y problemas identificados por el usuario:

Necesidades del sistema

1. Gestión de Diagnósticos de dispositivos tecnológicos
2. Gestión de Reparación y Mantenimientos.

Son los procesos principales que existe en el Departamento de Soporte Técnico de la empresa por lo cual son los requerimientos claves que debería tener el sistema.

#### **2.2.4 Procesamiento y Análisis de Datos**

Para poder obtener datos verídicos se ha empleado una encuesta a un total de 134 posibles usuarios del Sistema web a implementar en la empresa EPC Compu y de esta manera poder realizar un análisis de resultado. Así como también se ha decidido realizar una entrevista al personal de la empresa, quienes utilizaran el sistema con mayor frecuencia con el fin de comprobar que todos los requerimientos se hayan cumplido.

##### **2.2.4.1 Análisis e Interpretación de Resultados de la Encuesta Aplicada**

Mediante la encuesta realizada a un total de 134 personas siendo clientes de la empresa EPC Compu se ha podido evidenciar que en su gran mayoría opinan que el servicio es tardío o lento, y en estas ocasiones no conocen el motivo; posiblemente este sería uno de los mayores problemas por el cual los clientes tienden a dejar de utilizar el servicio o la empresa empieza a presentar problemas al momento de darle una respuesta efectiva al cliente.

Cabe recalcar que el resultado final del servicio lo consideran eficiente, el cual es un buen punto a favor de la empresa debido a que existe gran competencia en el mercado y las personas siguen considerando eficiente incluso teniendo ciertos inconvenientes.

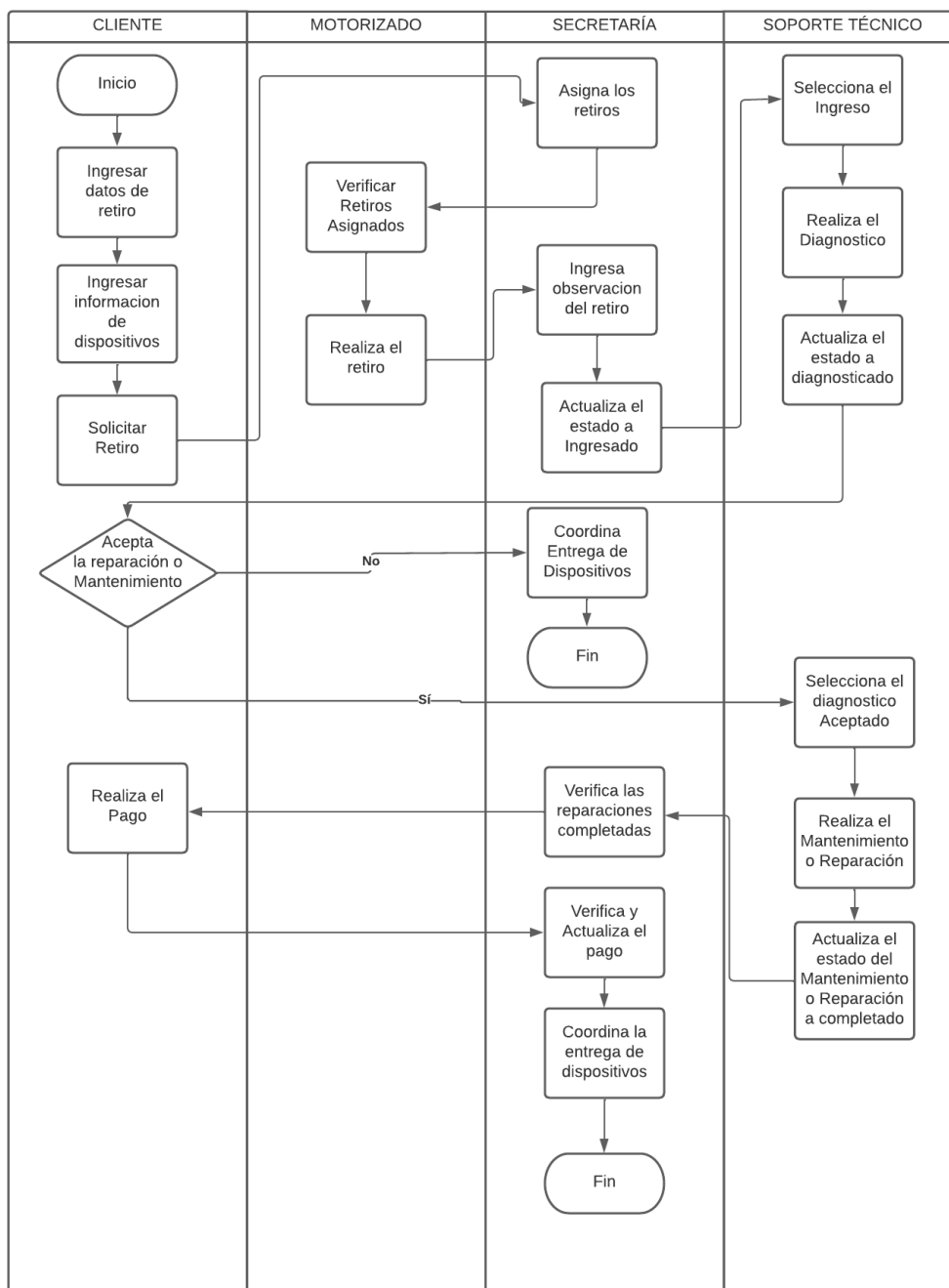
Se conoce que la gran mayoría de personas utilizan dispositivos como computadoras y laptops, que se debe aprovechar de manera oportuna y brindar un mejor servicio del que ya se ofrece y gracias a la encuesta se ha podido evidenciar que los clientes opinan de manera positiva a la creación de un sistema el cual se pueda visualizar el proceso de todo el servicio y de esa manera tener una idea más clara de todo lo que deben hacer antes de realizar el respectivo diagnóstico para ofrecer un servicio de calidad y mejorar la imagen de la empresa.



## CAPÍTULO III RESULTADOS Y DISCUSIONES

### 3.1 Análisis e Interpretación de resultados

#### 3.1.1 Descripción del Proceso de Soporte Técnico en la Empresa EPC Compu



**Figura 15.** Descripción del Proceso de Soporte Técnico en la Empresa EPC Compu  
**Elaborado por:** Valeria Quispe

### 3.1.2 Estudio Comparativo de Tecnologías de Front-End

#### 3.1.2.1 .NET Framework

Dicho framework tiene un conjunto de estructuras y varias tecnologías que son definidas para facilitar la programación, las cuales proporciona Microsoft teniendo en cuenta que dentro de .NET hasta la actualidad existe más de 20 lenguajes de programación. Proporciona un entorno de desarrollo que utiliza software simple, mejora la seguridad del programa y reduce las vulnerabilidades. La plataforma entró en el terreno de los servicios web, estableciendo XML (un lenguaje de marcado para el intercambio de información entre programas) como especificación o estándar. Cada versión de .NET Framework incluye algunas funciones de versiones anteriores y agrega nuevas funciones. En el framework .Net se incorpora el CLR (Common Language Runtime), que incluye los componentes base y algunos complementos como bibliotecas de clases base y bibliotecas de clases administradas. El CLR tiene su propia versión que no siempre coincide con la versión del marco .Net que se está utilizando.[26]

##### 3.1.2.1.1 Ventajas y Desventajas de .NET Framework

**Tabla 12.** Ventajas y Desventajas de .NET Framework  
**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
Programación Orientada a Objetos	Actualizaciones continuas, que en algunos momentos no se puede realizar los cambios de versiones anteriores.
El almacenamiento de cache es confiado y muy simple	Las aplicaciones deben seguir las reglas que tiene Microsoft para la realización de un proyecto
Utiliza un entorno de desarrollo integrado (IDE) como es Visual Studio	Dispone de una licencia libre con un uso bastante amplio, pero en el caso de que el proyecto sigue creciendo la licencia es muy costosa
Multiplataforma	Carece de un soporte cada vez que se actualiza una nueva versión

Dispone de un gran número de bibliotecas	
--	--

### 3.1.2.2 React

React puede ayudarlo a crear interfaces de usuario interactivas de una manera fácil. Diseñe vistas simples para cada estado en su aplicación, y React se encargará de actualizar y renderizar de manera eficiente los componentes correctos cuando cambien los datos. Las vistas declarativas hacen que su código sea más predecible y, por lo tanto, más fácil de depurar.

Los componentes de React implementan un método llamado `render()` que toma entradas y devuelve algo para mostrar. Este ejemplo usa una sintaxis similar a XML llamada JSX. Puede acceder a los datos de entrada pasados al componente a través de `render()` a través de `this.props`. [27]

#### 3.1.2.2.1 Ventajas y Desventajas de React

**Tabla 13.** Ventajas y Desventajas de React  
Elaborado por: Valeria Quispe

VENTAJAS	DESVENTAJAS
Tiene un gran nivel de flexibilidad	La documentación disponible no es oficial de React, por ende, los desarrolladores tienen dificultades.
La facilidad y sencillas de aprendizaje por tener una sintaxis muy simple.	Al momento de integración con interfaces requiere de un tiempo bastante prologando para dominarlo a un nivel master.
El framework es bastante ligero debido a que los datos del lado del usuario pueden representar en el servidor de manera simultánea.	Al momento de desarrollo tienen la dificultad de elegir un estándar.
La migración de versiones mucho más sencilla.	

### 3.1.2.3 Angular

Angular es un framework creado por Google de código abierto que hace la creación y programación de proyectos web mucho ms fácil debido a que son de una sola página. En este caso la plataforma separa por completo el frontend del backend para mantener todo su código de manera ordenada gracias a su patrón de diseño de software utilizado en este caso Modelo-Vista-Controlador (MVC), de esa manera permite un desarrollo ágil y rápido. [28]

#### 3.1.2.3.1 Ventajas y Desventajas de Angular

**Tabla 14.** Ventajas y Desventajas de Angular  
**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVANTAJAS</b>
Utilización de arquitectura MCV	Opciones limitadas de SEO
Hace más simple la manera de administración de componentes web.	Complejo y detallado
Servicios e inyección de dependencia (DI)	La curva de aprendizaje es más difícil en comparación con otras tecnologías.
Las directivas HTML que son personalizada mejoran su funcionalidad para que las aplicaciones sean más dinámicas al lado del cliente.	La interfaz de línea de comandos no tiene la información o documentación necesaria.
Debido a que está escrito con TypeScript que ayuda en la detección y eliminación de errores durante el proceso de a codificación.	

### 3.1.2.4 Cuadro Comparativo de Tecnologías Front-End

**Tabla 15.** Cuadro Comparativo de Tecnologías Front-End  
**Elaborado por:** Valeria Quispe

<b>CARACTERÍSTICAS</b>	<b>.NET FRAMEWORK</b>	<b>REACT</b>	<b>ANGULAR</b>
Tipo Aplicación	Aplicaciones enriquecidas (RIA)	Una página (SPA)	Una página (SPA)
Código Abierto	NO	SI	SI
Curva de Aprendizaje	MEDIA	Media	Media
DOM Virtual	SI	SI	NO
CREADOR	MICROSOFT	FACEBOOK	GOOGLE

### 3.1.2.5 Tecnología Front-End Elegida

Una vez que se ha realizado toda la investigación pertinente acerca de varias tecnologías Front-End se ha elegido .NET Framework gracias a diversas características y ventajas que ayudan al mejor desarrollo del presente proyecto, las cuales se las mencionan a continuación:

- La compilación en tiempo real y permite verificar paso a paso su código para verificar el fallo o el resultado en cualquier línea de código dentro del proyecto, de esa manera facilita el desarrollo del sistema web.
- Debido a que es un framework compatible con Visual Studio permite tener una mayor flexibilidad al añadir ciertos componentes que necesitan otro tipo de lenguaje front-end como HTML, Bootstrap y Java Script que mejora el desarrollo y el trabajo de la persona encargada de dicho desarrollo.

## 3.1.3 Estudio Comparativo de Tecnologías de Back-End

### 3.1.3.1 PHP

Lenguajes de código abierto utilizada para el desarrollo de aplicaciones web dinámicas, dentro del cual se puede añadir lenguaje HTML, al ser un lenguaje de

programación de código abierto permite a los desarrolladores flexibilizar la estructura y se pueda aplicar en cualquier tipo de proyecto.[29]

### 3.1.3.1.1 Ventajas y Desventajas de PHP

**Tabla 16.** Ventajas y Desventajas de PHP  
**Elaborado por:** Valeria Quispe

VENTAJAS	DESVENTAJAS
Código abierto	Configuraciones de seguridad dificultosas
Sintaxis amplia	No se puede ocultar el código fuente de manera fácil.
Flexible al momento de crear entornos de trabajo fácilmente	Necesariamente requiere de un servidor para su ejecución.
Integración simple con la base de datos	
Multiplataforma	

### 3.1.3.2 Python

Es un lenguaje de programación que durante los últimos años ha sido uno de los más utilizados por desarrolladores debido a sus características principales como son la legibilidad con la que se puede realizar cualquier proyecto siendo fácil de entender y programar, gracias a su gran diversidad de bibliotecas son un gran apoyo para evitar escribir cada código de forma manual. [30]

### 3.1.3.2.1 Ventajas y Desventajas de Python

**Tabla 17.** Ventajas y Desventajas de Python  
**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVANTAJAS</b>
Código abierto y gratuito	Interrupciones hacen que se ejecuten los programas de manera lenta
Función de programación que ayudan a la creación de proyectos IoT	Para acceder a la base de datos es un poco tedioso debido a que no están completamente desarrolladas
Se lo puede integrar en varias opciones	Se debe realizar una gran cantidad de pruebas para detectar errores
Bibliotecas extensas de gran ayuda	Depende de ciertas bibliotecas

### 3.1.3.3 ASP.NET

ASP.NET permite desarrollar diferentes tipos de aplicaciones en la Web, desde sitios con pequeñas implementaciones hasta grandes sistemas preparados para soportar mucho trabajo y operaciones al mismo tiempo. Hay muy pocos requisitos para comenzar a aprender y desarrollar aplicaciones en ASP.NET. Además de los conocimientos previos que necesitas, todo lo que necesitas es una computadora con sistema operativo Windows o Mac, acceso a internet y un editor de código. Puede usar cualquier editor de código o IDE, como Visual Studio Community. Este es un IDE creado por Microsoft que está completamente integrado con la plataforma de desarrollo ASP.NET. Es gratis y en la versión mencionada te permite desarrollar cómodamente con herramientas como detección de errores de sintaxis, scaffolding, pruebas de aplicaciones y herramientas de arranque, manejo de base de datos, e incluso con versionado de código.[31]

#### 3.1.3.3.1 Ventajas y Desventajas de ASP.NET

**Tabla 18.** Ventajas y Desventajas de ASP.NET

**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
Soporte técnico actualizado por la comunidad de Microsoft	Velocidad del sistema con menor rendimiento en comparación con Linux
Se puede combinar con varios lenguajes	Necesita una plataforma específica para el desarrollo
Arquitectura MVC	Problemas de compatibilidad con sistemas operativos distintos a Windows
Creación de aplicaciones web mediante formularios	
Estabilidad en la conexión con SQL Server	

### 3.1.3.4 Cuadro Comparativo de Tecnologías Back-End

**Tabla 19.** Cuadro Comparativo de Tecnologías Back-End

**Elaborado por:** Valeria Quispe

<b>CARACTERÍSTICAS</b>	<b>PHP</b>	<b>PHYTON</b>	<b>ASP .NET</b>
Aprendizaje	Curva de aprendizaje Medio	Curva de aprendizaje Fácil	Curva de aprendizaje Fácil
Rendimiento	Genera Alto Rendimiento	Su rendimiento es Medio en cuanto a implementación	Su rendimiento es Medio en cuanto a implementación
Flexibilidad	La integración con HTML, XML y otros códigos	Adaptación de forma simple	Integración con varios lenguajes mediante librerías
Compatibilidad	Casi todos los sistemas operativos	Casi todos los sistemas operativos	Casi todos los sistemas operativos
Costes	Open Source	Open Source	Pagado



### **3.1.3.5 Tecnología Back-End Elegida**

De acuerdo con la investigación realizada durante la realización del proyecto entre las tecnologías Back-End se ha elegido la mejor opción ASP.NET debido a sus grandes beneficios para el desarrollo de aplicaciones web.

- ASP.NET le brinda todo lo que necesita para desarrollar aplicaciones web, redes sociales, sitios web comerciales y sistemas de administración de contenido (CMS). Esto se debe a que te permite combinar su marco de desarrollo con XML, PHP y HTML para que puedas crear con éxito sitios web interactivos sin un mantenimiento complicado.
- Es un marco de desarrollo que le permite crear aplicaciones web simples en páginas. Tiene la ventaja de que puede desarrollar su código en la misma interfaz en la que se verá el producto final.
- Utiliza controladores de vista de modelo, una arquitectura que separa la parte de desarrollo web de la interfaz de diseño. Esto le brinda más flexibilidad cuando trabaja en equipo, ya que algunos desarrollan código y otros se dedican al diseño sin romper ningún código.

### **3.1.4 Metodología para el Desarrollo del Software**

La metodología de desarrollo de software es un conjunto de técnicas y métodos organizativos utilizados para diseñar soluciones de software informático. El objetivo de los diferentes enfoques es tratar de organizar los equipos de trabajo para que desarrollen la funcionalidad del programa de la mejor manera posible.

#### **3.1.4.1 Waterfall o Cascada**

Es una de las metodologías tradicionales más utilizadas, en el cual se debe realizar el desarrollo del proyecto de manera secuencial para poder ir por fases realizando el proyecto [32]. Está compuesta por 7 fases que son de gran importancia, las cuales pueden mencionar:

1. Captura y documentación de requisitos
2. Diseño

3. Desarrollo
4. Test
5. UATs
6. Corrección de errores y ajustes finales
7. Puesta en producción

#### 3.1.4.1.1 Ventajas y Desventajas de Waterfall o Cascada

**Tabla 20.** Ventajas y Desventajas de Waterfall o Cascada  
**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
Llegan a un acuerdo en una de las fases más tempranas del desarrollo del proyecto.	Es tedioso al momento de recabar requerimientos en conjunto con el cliente debido a que muchas de las veces existen requerimientos muy específicos
Se puede medir el progreso del desarrollo de manera más fácil y eficiente.	Existe una leve posibilidad que al final el cliente no quede satisfecho con el producto entregable.
No se requiere la presencia del cliente en todas las fases.	
El personal encargado de ciertos procesos pueden ser parte de estos y de otros proyectos o fases.	

#### 3.1.4.2 Scrum

Ayuda al trabajo colaborativo entre equipos de trabajo para que logren un producto deseado con la mayor organización posible en el cual se va aprendiendo según las experiencias y la autoorganización para poder solventar un problema durante el proceso de desarrollo.

Marco de trabajo

El marco de trabajo que tiene la metodología Scrum, tiene como base el aprendizaje continuo y sobre todo adaptarse fácilmente a los distintos factores que conlleva el desarrollo de un proyecto, se lo ha diseñado para que el equipo de trabajo se adapte de mejor manera a las condiciones que irán cambiando a lo largo del desarrollo y sobre todo a los requerimientos de los usuarios finales.

#### Artefactos de Scrum

##### Backlog del producto

Hace referencia al listado de todas las tareas o requerimientos que se va a realizar en el proyecto. Se lo mencionaría coloquialmente como las cosas que están por hacer para que el equipo siempre lo pueda revisar y verificar cada avance.

##### Backlog de sprint

En este caso se podría hablar de las historias de usuario que se ha seleccionado por equipo de desarrollo o por personal. Para poder presentar dichos sprint se debe realizar una reunión en la cual se explicará de manera detallada las tareas que se realizarán durante el desarrollo del proyecto.

##### Incremento

Se verifica cada sprint terminado y se lo presenta a todo el equipo de trabajo.

##### Protocolos o eventos de scrum

Scrum tiene varios eventos que son muy importantes a la hora de realizar un proyecto de desarrollo y de esa manera tener la menor cantidad de errores al final. Entre ellos podemos mencionar los siguientes protocolos:

1. Organización del backlog
2. Planificación de sprint
3. Sprint
4. Scrum diario o reunión rápida
5. Revisión del sprint
6. Retrospectiva de sprint [33]

### 3.1.4.2.1 Ventajas y Desventajas de Scrum

**Tabla 21.** Ventajas y Desventajas de Scrum  
**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
Flexibilidad con el grupo de trabajo y las etapas del proceso debido a que pueden trabajar de manera conjunta y proponer soluciones antes problemas existentes.	Ayuda de mejor manera a equipos pequeños de desarrollo.
Durante cada una de las etapas de desarrollo, nos da ya un resultado y el cliente no debe esperar hasta finalizar el proyecto	Se debe definir específicamente todas las tareas a realizar con sus respectivos tiempos.
Adaptación y flexibilidad con cualquier tipo de proyecto.	Se necesita tener un amplio conocimiento en el tema debido a que no es apto para desarrolladores junior.
Los errores durante el desarrollo se van gestionando en el momento que aparecen y le dan una breve solución.	

### 3.1.4.3 Kanban

Ayuda a mejorar la organización de todas las tareas que conlleva un proyecto de manera eficiente y efectiva hasta la finalización de todo el trabajo. La metodología KanBan hace referencia a tarjetas visuales traducido del japonés. Para poner en práctica toda la metodología se debe crear un tablero de tareas en el cual se puede organizar todo el trabajo según los requerimientos.

#### **Funcionalidad de Kanban**

Es una de las más sencillas de utilizar en el cual se puede actualizar los estados de cada proceso, de esa manera realizar un análisis de los avances del proyecto y en el caso de alguien ha terminado sus tareas poder asignarle nuevas tareas o a nuevos proyectos.

### **1. Definir el flujo de trabajo por proyecto**

Se crea un tablero al cual todos los miembros del equipo tengan acceso, para que de esa manera mientras siguen avanzando se vaya proyectando en el tablero.

### **2. Fases del ciclo de producción**

Todas las tareas que conlleva el proyecto de lo debe colocar en una tarjeta que describa todo lo que se va a realizar y el personal responsable. En el cual se irán colocando los procesos y observaciones que tengan con las tareas para que puedan ir cambiando el estado de cada tarea.

### **3. Stop starting, start finishing**

Se debe priorizar tareas de acuerdo como avance el proyecto para poder solventar las nuevas tareas que aparecen durante el proceso de producción, para que esto no sea un inconveniente se deben limitar el número de tareas por cada fase.

### **4. Control de Flujo**

Se realiza un seguimiento a las tareas y proyecto por hacer para verificar cuál de ellas puede tener mayor inconveniente y se pueda solventar en grupo.

### 3.1.4.3.1 Ventajas y Desventajas de Kanban

**Tabla 22.** Ventajas y Desventajas de Kanban  
**Elaborado por:** Valeria Quispe

<b>VENTAJAS</b>	<b>DESVENTAJAS</b>
Se puede medir el rendimiento del equipo de trabajo.	La versión gratuita tiene un límite de tareas, en caso de ser un proyecto muy grande el costo será muy elevado.
Tiene una mayor organización de trabajo.	Se lo utiliza más con equipos de trabajo reducidos.
Cumplir con los tiempos estimados de entrega.	Específico para ciertos proyectos debido a las variaciones durante el proceso.
Distribución de tareas más eficiente dentro del equipo de trabajo.	No se debe anticipar a la carga de trabajo más grande debido a la acumulación de tareas y sin poder medir el proceso.

### 3.1.4.4 Cuadro Comparativo de Metodologías de Desarrollo

**Tabla 23.** Cuadro Comparativo de Metodologías de Desarrollo  
**Elaborado por:** Valeria Quispe

CARACTERÍSTICAS	CASCADA	KANBAN	SCRUM
Funciones	Equipo de desarrollo	Sin funciones definidas	Equipo de desarrollo, máster scrum
Cadencia/Cronograma	El equipo puede pasar a la siguiente fase sólo cuando se haya completado la anterior	Trabajo fluye de acuerdo con las necesidades	Depende de la duración del sprint
Metodología de Lanzamiento	Planificación proyectos con anticipación	Entrega Continua	Valor entregado al final de cada sprint
Métricas primarias	Proceso lineal	Tiempo del ciclo, plazo de entrega, cantidad de trabajo de proceso	La velocidad y el valor creados
Modificación y Cambios	Ofrece la oportunidad de detectar la fase del problema en la que surgió	Los cambios se realizan en cualquier momento	Determinados y aplicados entre los sprints

### 3.1.4.5 Metodología de Desarrollo Elegida

Durante la investigación de las distintas metodologías para desarrollo de software se ha elegido como la óptima para el proyecto en este caso sería metodología Kanban al ser un proyecto para una empresa pequeña y no requieren de muchos requerimientos adicionales a los que se ha presentado en puntos anteriores.

Dicha metodología tiene una serie de ventajas que nos ayudaran a mejorar el proyecto durante el proceso, entre las que podemos mencionar:

- Gestión de trabajo fluida
- Medición del equipo de trabajo
- Organización para la realización de tareas
- Asignación de tareas
- Entrega un producto de calidad

### **3.2 Desarrollo de la Propuesta**

### **3.3 Aplicación de la Metodología**

Se ha seleccionado como mejor metodología Kanban, debido a que será de gran ayuda para la realización del presente proyecto, y así a través de las distintas fases que debe cumplir se llegue a entregar un producto de calidad al cliente final.

Dentro de la metodología Kanban existe cinco prácticas para la dirección de proyectos, los cuales se mencionan a continuación:

#### **Visualización del flujo de trabajo**

Se debe realizar un análisis de todas las tareas que conlleva el proyecto de desarrollo y el equipo de trabajo que se va a utilizar para la realización de este.

#### **Limitar el WIP (Work in Progress)**

Realiza el control de todo el trabajo que se debe realizar en el proyecto para que de esa manera se pueda ordenar todo el trabajo y realizar una entrega a tiempo y en óptimas condiciones.

#### **Gestión del flujo**

Se identifican cuellos de botella o inconvenientes durante el proceso de desarrollo para mejorarlo y tener una mejor fluidez.

#### **Políticas Explicitas**



Se realiza la documentación de todos los procedimientos de trabajo para garantizar que se estén ejecutando de forma transparente de acorde a lo acordado con el cliente.

#### **Evolución empírica y mejora colaborativa**

Se establecen pruebas durante la realización de las tareas para tener un flujo de trabajo efectivo y eficiente de acuerdo con los tiempos de entrega y la calidad del producto final.

#### **Implementación de ciclos de comentarios**

Es necesario tener el punto de vista tanto del cliente como del equipo de trabajo para recabar más información y obtener mejor resultados para ambas partes. Y como punto importante ayuda a fomentar el liderazgo de un grupo de trabajo.

### **3.3.1 Arquitectura de la Aplicación**

### 3.3.2 Visualizar el flujo de Trabajo

Durante la realización del presente proyecto, se generaron distintos requerimientos dentro de la empresa EPC Compu, para lo cual se realizaron historias de usuario que se van a ejecutar durante todo el proceso y posteriormente realizar todas las pruebas respectivas dentro de la empresa.

**Tabla 24.** Diseño de la base de Datos  
**Elaborado por:** Valeria Quispe

<b>Historia de Usuario</b>	
<b>Número:</b> HU01	<b>Usuario:</b> Desarrollador
<b>Nombre de la Historia:</b> Diseño de la base de Datos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 5	<b>Iteración Asignada:</b> 1
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
La creación de la base de datos se lo realizara tomando en cuenta los distintos requerimientos solicitados por la empresa y aquellos que surgieron del tratamiento de datos obtenidos en la fase de recolección de información.	
<b>Observación</b>	
El usuario podrá tener una vista clara y manejable de la base de datos identificando de manera concisa las entidades y sus respectivas relaciones.	

**Tabla 25.** Diseño de la página principal del Sistema de Gestión de Soporte Técnico  
**Elaborado por:** Valeria Quispe

<b>Historia de Usuario</b>	
<b>Número:</b> HU02	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Diseño de la página principal del Sistema de Gestión de Soporte Técnico	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 4	<b>Iteración Asignada:</b> 2
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Mediante la página principal del sistema se dará una breve descripción de la empresa EPC Compu a los distintos clientes para que de esa manera tengan una idea más clara de los servicios que ofrece dicha empresa.	
<b>Observación</b>	
El usuario podrá visualizar y navegar por las distintas opciones como Servicios, Blog de Noticias y Galería de Fotos que ha puesto como evidencia la empresa, y de esa manera tiene la opción de ingresar al sistema.	

**Tabla 26.** Creación de nuevos usuarios  
**Elaborado por:** Valeria Quispe

<b>Historia de Usuario</b>	
<b>Número:</b> HU03	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Creación de nuevos usuarios.	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 4	<b>Iteración Asignada:</b> 2
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
La página permite al usuario crear una cuenta nueva mediante el llenado de un formulario básico de registro para que posteriormente le permitirá ingresar al sistema.	
<b>Observación</b>	
Los datos proporcionados en el formulario quedaran guardados en el sistema para que el usuario ya no se registre por varias ocasiones con un mismo correo electrónico.	

**Tabla 27. Login del sistema**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU04	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Login del sistema.	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 4	<b>Iteración Asignada:</b> 2
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
La página permite al usuario colocar sus credenciales como son correo electrónico y contraseña para poder ingresar al sistema	
<b>Observación</b>	
Las credenciales en el ingreso al sistema son validadas por el sistema con los datos que ingreso al crear su cuenta.	

**Tabla 28. Página principal de Ingreso al Sistema**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU05	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Página principal de Ingreso al Sistema	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 2
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Inicialmente la empresa siempre te brindará una cordial bienvenida a su sistema web en el cual podrá navegar en cada una de sus opciones de menú	
<b>Observación</b>	
El usuario puede realizar las distintas tareas dentro del sistema iniciando por la generación del retiro de sus equipos	

**Tabla 29. Gestión de Retiros**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU06	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Gestión de Retiros	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 3
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
<p>El usuario podrá visualizar todo lo que conlleva la gestión de Retiros dentro del sistema como son:</p> <ul style="list-style-type: none"> <li>• <b>Solicitar Retiros</b> El usuario solicitará el servicio de retiro detallando su información personal, dirección de domicilio y el ingreso de los dispositivos correspondientes.</li> <li>• <b>Estado de Retiros</b> El usuario podrá ver un historial de los estados de su respectivo retiro para verificar el proceso en el cual se encuentra.</li> <li>• <b>Asignar Retiros</b> El usuario podrá visualizar y asignar al personal encargado los retiros pendientes.</li> <li>• <b>Ver Asignaciones de Retiros</b> El usuario va a visualizar los retiros asignados para recolección.</li> <li>• <b>Ingresar Retiros</b> El usuario realizará el ingreso de los dispositivos a la empresa.</li> </ul>	
<b>Observación</b>	
<p>Se permitirá el ingreso y actualización de recorridos como el cambio de estado de acuerdo con su proceso de la siguiente manera:</p> <ul style="list-style-type: none"> <li>• <b>Iniciado:</b> Cliente solicito el retiro a domicilio de los dispositivos</li> <li>• <b>Proceso:</b> Personal de la empresa se comunicará con el cliente para realizar el retiro en el domicilio</li> <li>• <b>Ingresado:</b> Personal de la empresa receptara los dispositivos recogidos del domicilio</li> <li>• <b>Revisión:</b> Personal técnico se encuentra en revisión de los equipos tecnológicos</li> <li>• <b>Diagnosticado:</b> Personal de la empresa emitirá la orden de Reparación o Mantenimiento al cliente</li> <li>• <b>Completado:</b> El cliente aceptará o rechazará realizar el trabajo, empezando el proceso de reparación</li> </ul>	

**Tabla 30. Menú de Retiros**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU07	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Menú de Retiros	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 3
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Se mostrará al usuario la gestión de los retiros para su debido uso.	
<b>Observación</b>	
Se le dará la información necesaria al usuario para que pueda manejar toda la parte de gestión de retiros dentro del sistema web.	

**Tabla 31. Gestión de Diagnósticos**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU08	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Gestión de Diagnósticos	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 4
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
<p>El usuario podrá visualizar todo lo que conlleva la gestión de Retiros dentro del sistema como son:</p> <ul style="list-style-type: none"> <li>• <b>Seleccionar Ingreso</b> El usuario podrá seleccionar uno de los ingresos registrados en el local para su respectivo proceso.</li> <li>• <b>Ingresar Diagnóstico</b> Una vez el personal de soporte técnico ha realizado el respectivo diagnóstico lo ingresa al sistema.</li> <li>• <b>Aceptar Diagnóstico</b> El usuario verifica el reporte del diagnóstico realizado a los dispositivos para aceptar o rechazar la reparación o mantenimiento.</li> <li>• <b>Ver Diagnóstico</b> El usuario podrá visualizar el estado del diagnóstico.</li> </ul>	
<b>Observación</b>	
<p>Se permitirá al usuario el ingreso y actualización de diagnóstico como su cambio de estado de acuerdo con el punto en el cual se encuentre, y se lo va a expresar de la siguiente manera:</p> <ul style="list-style-type: none"> <li>• <b>Aceptado</b> Una vez aceptado el diagnóstico se procederá a realizar la reparación o mantenimiento de los dispositivos, conforme al precio establecido y se podrá visualizar el estado de este a través del sistema</li> <li>• <b>Rechazado</b> Una vez rechazado el diagnostico se finalizará el proceso, informando al usuario el acercamiento para el retiro de los dispositivos.</li> </ul>	

**Tabla 32. Menú de Diagnósticos**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU09	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Menú de Diagnósticos	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 4
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Se mostrará al usuario la gestión de los diagnósticos para su debido uso.	
<b>Observación</b>	
Se le dará la información necesaria al usuario para que pueda manejar toda la parte de gestión de diagnósticos dentro del sistema web.	

**Tabla 33. Gestión de Reparación – Mantenimiento**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU10	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Gestión de Reparación - Mantenimiento	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 5
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
<p>El usuario podrá visualizar todo lo que conlleva la gestión de Reparación-Mantenimiento dentro del sistema como son:</p> <ul style="list-style-type: none"> <li>• <b>Seleccionar diagnostico</b> El personal de soporte técnico elige un diagnóstico con previa aceptación del usuario para proceder a su respectiva reparación o mantenimiento.</li> <li>• <b>Finalizar Reparación</b></li> <li>• <b>Estado Reparación</b></li> </ul>	
<b>Observación</b>	
Se permitirá al usuario el ingreso y actualización de Reparación-Mantenimiento como su cambio de estado de acuerdo con el punto en el cual se encuentre, y se lo va a expresar de la siguiente manera:	



**Tabla 34. Menú de Reparación-Mantenimiento**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU11	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Menú de Reparación-Mantenimiento	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 5
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Se mostrará al usuario la gestión de la Reparación-Mantenimiento para su debido uso.	
<b>Observación</b>	
Se le dará la información necesaria al usuario para que pueda manejar toda la parte de gestión de Reparación-Mantenimiento dentro del sistema web.	

**Tabla 35. Gestión de Reportes**  
**Elaborado por: Valeria Quispe**

<b>Historia de Usuario</b>	
<b>Número:</b> HU12	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Gestión de Reportes	
<b>Prioridad en Negocio:</b> Alta	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 6
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
El usuario podrá visualizar todos los reportes durante el proceso de servicio técnico para que tenga un mejor detalle de todo lo que se está haciendo con los dispositivos enviados a revisión.	
<b>Observación</b>	
Mediante reportes todos los usuarios del sistema tendrán un mayor conocimiento e información ordenada y organizada.	

**Tabla 36.** Gestión de Usuarios  
**Elaborado por:** Valeria Quispe

<b>Historia de Usuario</b>	
<b>Número:</b> HU13	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Gestión de Usuarios	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 7
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Se realizará un listado de roles de usuario para poder realizar toda la gestión necesaria dentro del sistema y tenga los permisos adecuados dependiendo el tipo de usuario.	
<b>Observación</b>	
Mediante los roles de usuario, cada uno de ellos tendrá acceso únicamente a parte del sistema que sean necesarias.	

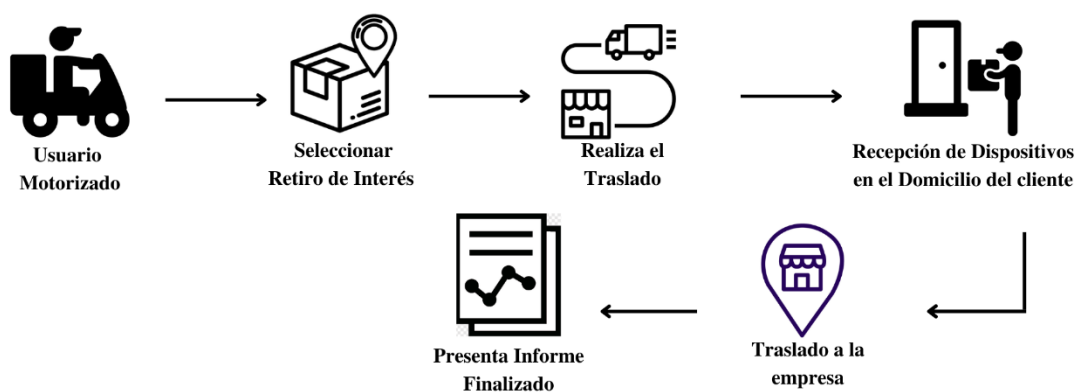
**Tabla 37.** Acceso al menú por tipo de usuario  
**Elaborado por:** Valeria Quispe

<b>Historia de Usuario</b>	
<b>Número:</b> HU14	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Acceso al menú por tipo de usuario	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 7
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
Según el rol de usuario que cumple cada usuario tiene acceso a un menú en específico en el cual podrá manejar el sistema.	
<b>Observación</b>	
El menú que disponga el usuario será adecuado de acuerdo con su rol dentro del sistema	

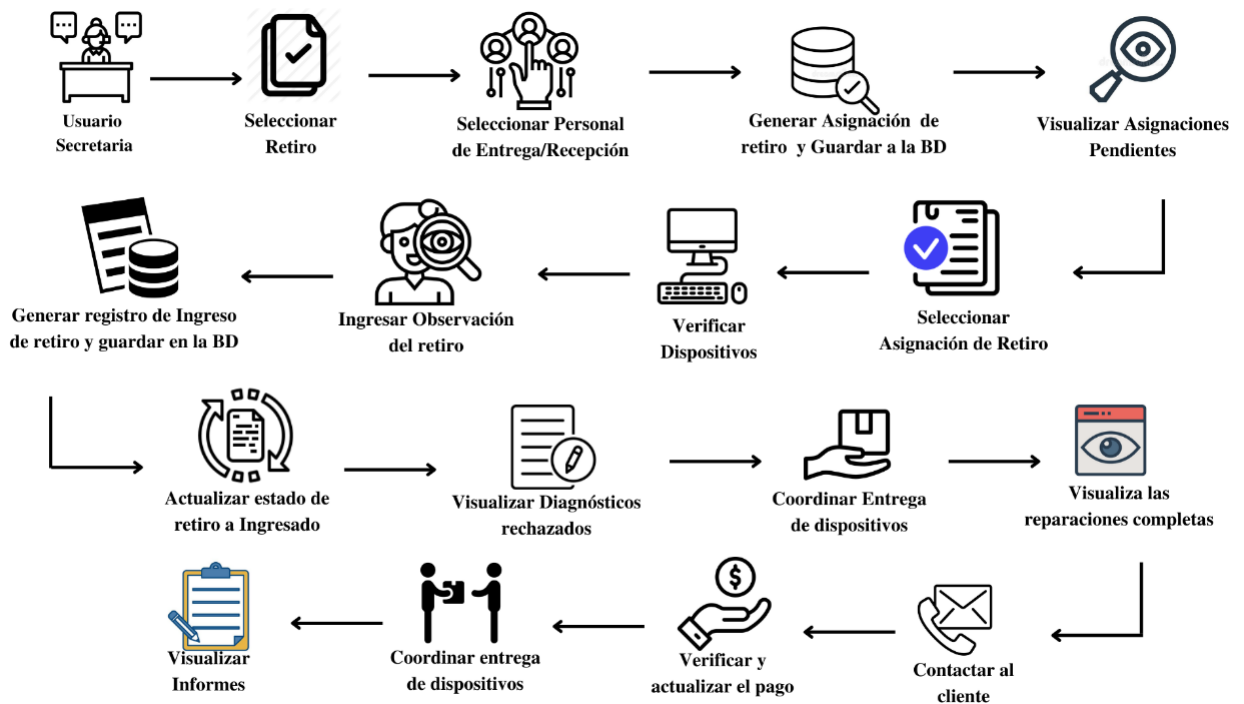
**Tabla 38.** Manejo del Perfil  
**Elaborado por:** Valeria Quispe

<b>Historia de Usuario</b>	
<b>Número:</b> HU15	<b>Usuario:</b> Cliente
<b>Nombre de la Historia:</b> Manejo del Perfil	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo de Desarrollo:</b> Alta
<b>Puntos Estimados:</b> 2	<b>Iteración Asignada:</b> 8
<b>Programador Responsable:</b> Valeria Quispe	
<b>Descripción</b>	
El usuario podrá cambiar parte de su información personal mediante el sistema en caso de que lo necesite o lo requiera.	
<b>Observación</b>	
La actualización de información o datos personales es de suma importancia debido a que se utilizara para comunicarse desde la empresa.	

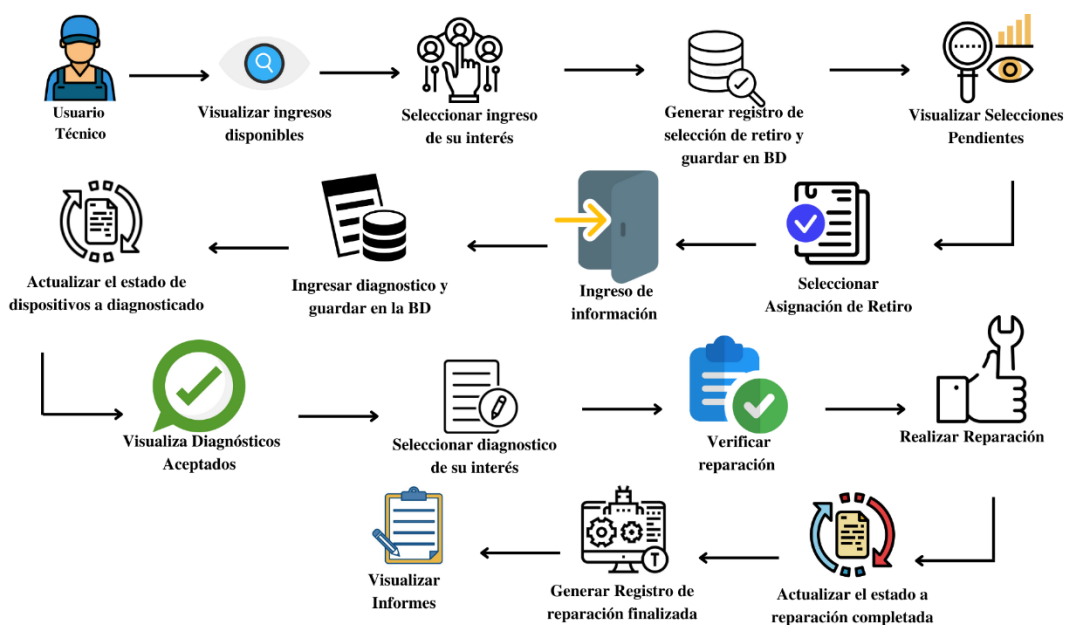
Al finalizar la descripción de cada historia de usuario, es mucho más entendible visualizar el flujo de trabajo para la realización del presente proyecto y de esa manera entregar un producto de calidad al cliente. A continuación, se muestra los diagramas del sistema por tipo de usuario.



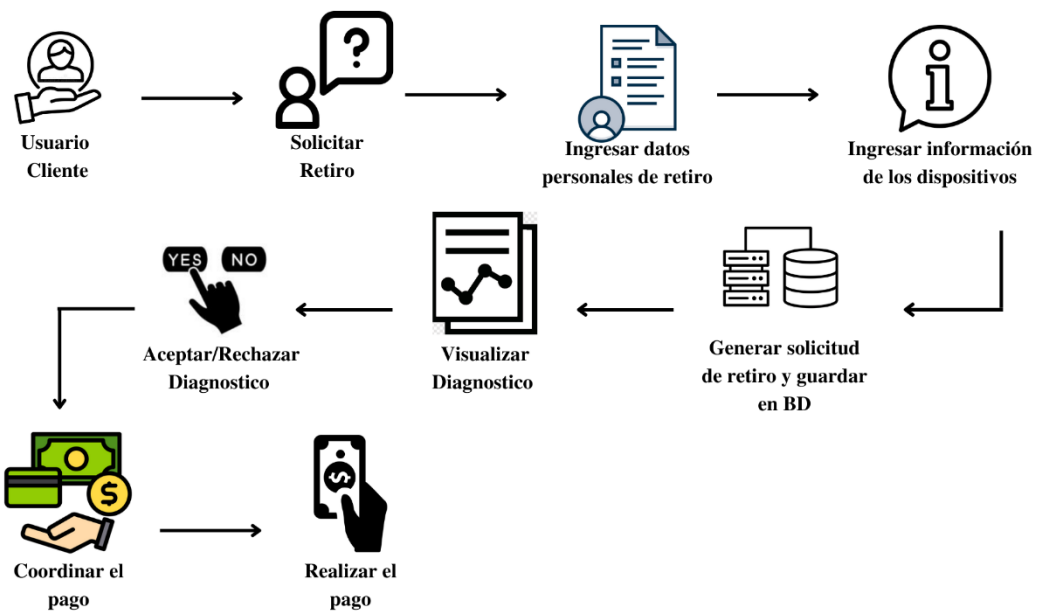
**Figura 16.** Diagrama de Flujo del Sistema- Usuario Motorizado  
**Elaborado por:** Valeria Quispe



**Figura 18.** Diagrama de Flujo del Sistema- Usuario Secretaria  
**Elaborado por:** Valeria Quispe



**Figura 17.** Diagrama de Flujo del Sistema- Usuario Técnico  
**Elaborado por:** Valeria Quispe



**Figura 19.** Diagrama de Flujo del Sistema- Usuario  
**Elaborado por:** Valeria Quispe

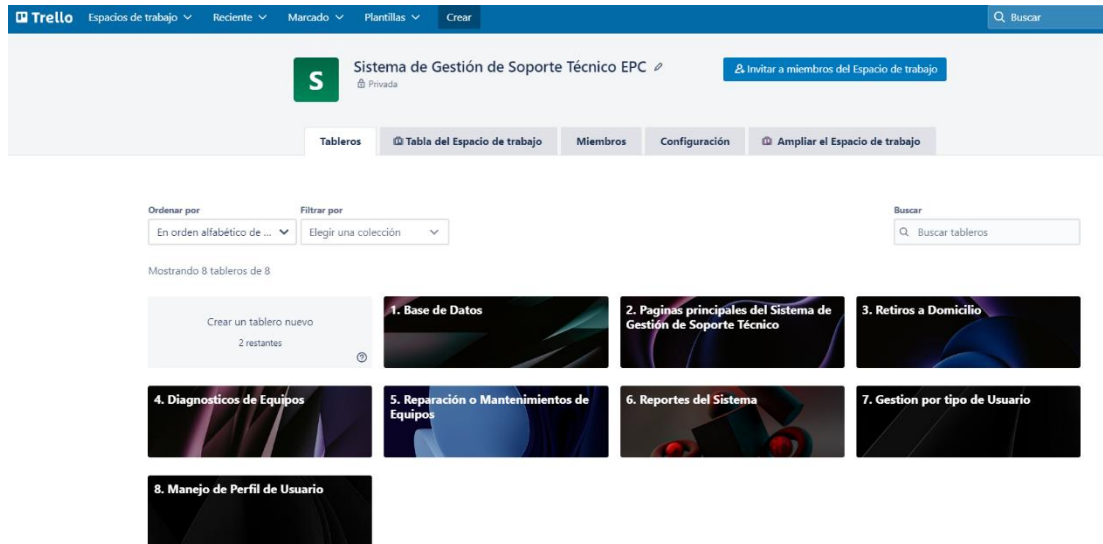
### 3.3.3 Limitar la cantidad de trabajo (WIP)

Dentro de la metodología que se ha elegido tiene un punto muy importante que es el WIP o Work in Progress, que ayuda a mejorar la organización de un equipo de trabajo, debido a que se reparte ciertas tareas de acuerdo a la capacidad de cada integrante para que lo pueda realizar de manera más fácil.

Para la aplicación de la metodología Kanban se ha utilizado la herramienta Trello, que permite una mejor organización de los procesos que requiere el proyecto, para que de esa manera se pueda limitar ciertas tareas y poder revisar en conjunto con el cliente en este caso la empresa EPC Compu. Cada iteración va a tener sus respectivas historias de usuario las cuales incluyen ciertas tareas.

### 3.3.3.1.1 Requerimientos del Sistema

Mediante la creación de un proyecto en Trello se ha determinado las iteraciones como tableros de trabajo para realizar las tareas y subtareas correspondientes al proyecto, de esa manera se mantendrá un orden específico al momento de ir completando cada proceso. Y así evitaremos cualquier tipo de inconveniente incluyendo cuellos de botella.



**Figura 20.** Creación de proyecto herramienta Trello para Kanban

**Elaborado por:** Valeria Quispe

### 3.3.3.1.2 Lista de tareas por hacer

Cada tablero lleva las historias de usuario correspondiente a la iteración, dentro de los cuales existen una lista de tareas que se deben ir completando en el transcurso del desarrollo del proyecto.

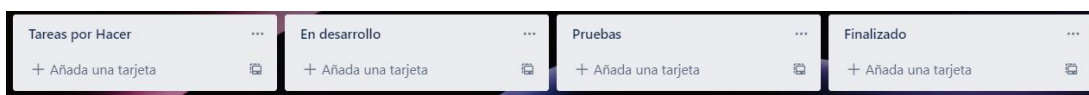


**Figura 21.** Lista de tareas a realizar en el presente proyecto

**Elaborado por:** Valeria Quispe

### 3.3.4 Flujo de Trabajo

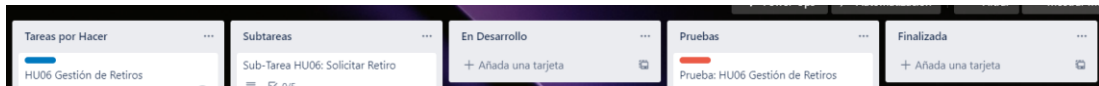
Se ha determinado que el flujo de trabajo tendrá varios aspectos importantes como son, las **tareas por hacer** que son la lista de cosas pendientes, en **desarrollo** cuando las tareas están en proceso, **pruebas** las cuales se las realizaran una vez terminada el desarrollo y finalmente si pasan las pruebas la tarea de da por **finalizada**.



**Figura 22.** Flujo de trabajo metodología Kanban

**Elaborado por:** Valeria Quispe

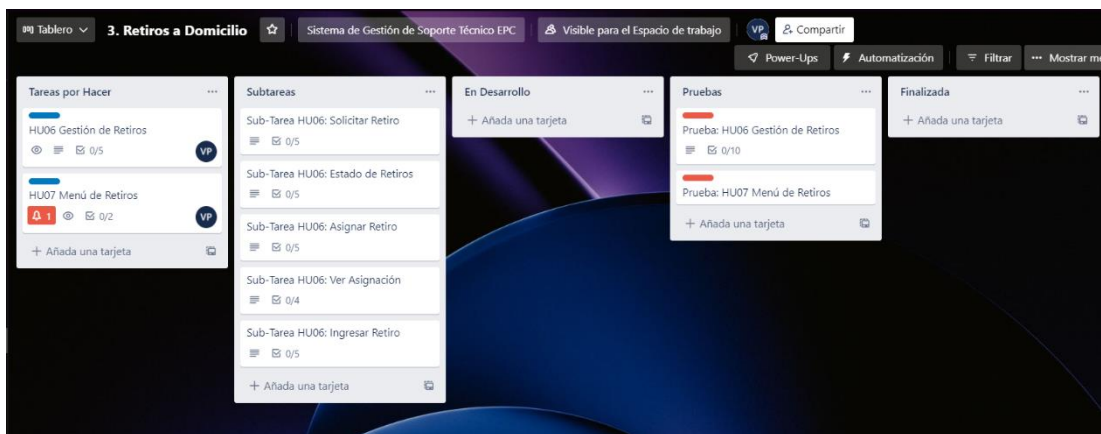
En ciertas ocasiones dependiendo las tareas se van a subdividir en subtareas que necesitan pasar por el mismo proceso que una tarea el cual se ha creado una nueva lista dentro del flujo de trabajo que lo visualizara de la siguiente manera.



**Figura 23.** Subtareas por realizar partiendo de la metodología Kanban  
**Elaborado por:** Valeria Quispe

### 3.3.5 Políticas Explícitas

Se identifica que todo el proceso de desarrollo del producto se lleva de manera ética y transparente por parte de la persona encargada para que el cliente pueda identificar que se lo hizo de acuerdo con todos los requerimientos acordados



**Figura 24.** Políticas explícitas, parte de la metodología Kanban  
**Elaborado por:** Valeria Quispe

### 3.3.6 Desarrollo de Tareas

#### HU01 Diseño de la Base de Datos

El diseño de la base de datos se encuentra enfocado en la comprensión de los diferentes procesos que maneja la empresa encargada de realizar la reparación y mantenimiento de dispositivos tecnológicos, con ello se prevé realizar la identificación de entidades inmersas con sus respectivas relaciones.





**Figura 25.** Tareas y Subtareas del Diseño de la Base de Datos  
**Elaborado por:** Valeria Quispe

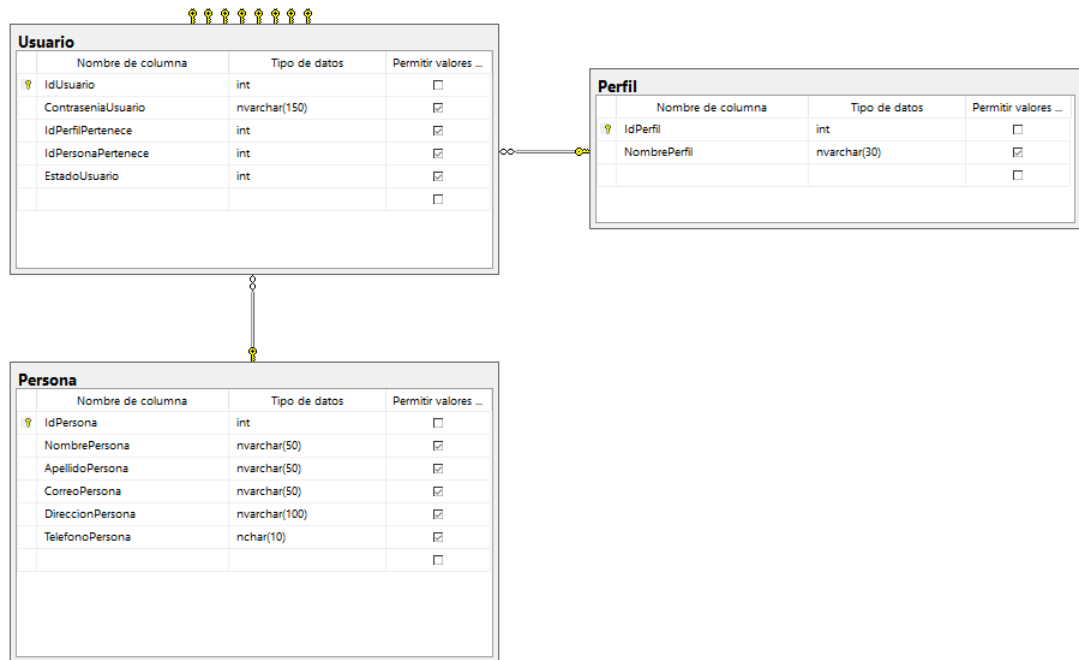
### Explicación

Conforme al análisis se identificó las diferentes entidades con sus respectivos atributos como:

- + [db] dbo.AsignacionRetiro
- + [db] dbo.DiagnosticoCabecera
- + [db] dbo.DiagnosticoDetalle
- + [db] dbo.EstadoReparacion
- + [db] dbo.EstadoRetiro
- + [db] dbo.IngresoRetiro
- + [db] dbo.Perfil
- + [db] dbo.Persona
- + [db] dbo.Reparacion\_Mantenimiento
- + [db] dbo.ReparacionCompleta
- + [db] dbo.RetiroCabecera
- + [db] dbo.RetiroDetalle
- + [db] dbo.SeleccionIngreso
- + [db] dbo.Usuario

**Figura 26.** Entidades de la Base de Datos  
**Elaborado por:** Valeria Quispe

El desarrollo de las relaciones se lo realizó a través del modelo entidad-relación, mismo que fue llevado a cabo en el gestor de base de datos SQL SERVER, para ello se generaron las respectivas relaciones entre las diferentes entidades con el fin de mantener la integridad de las operaciones.



**Figura 27.** Relaciones de la Base de Datos  
Elaborado por: Valeria Quispe

## Pruebas

El fin de las pruebas es conllevar de forma correcta las claves primarias y foráneas es decir en manejar de manera adecuada las relaciones existentes.

**Prueba: HU01 Diseño de la Base de Datos**  
en la lista [Pruebas](#)

Etiquetas  
**Prueba** +

Descripción [Editar](#)

Verificar las entidades con sus respectivas relaciones, previo anexar un diagrama del modelo entidad relación del sistema

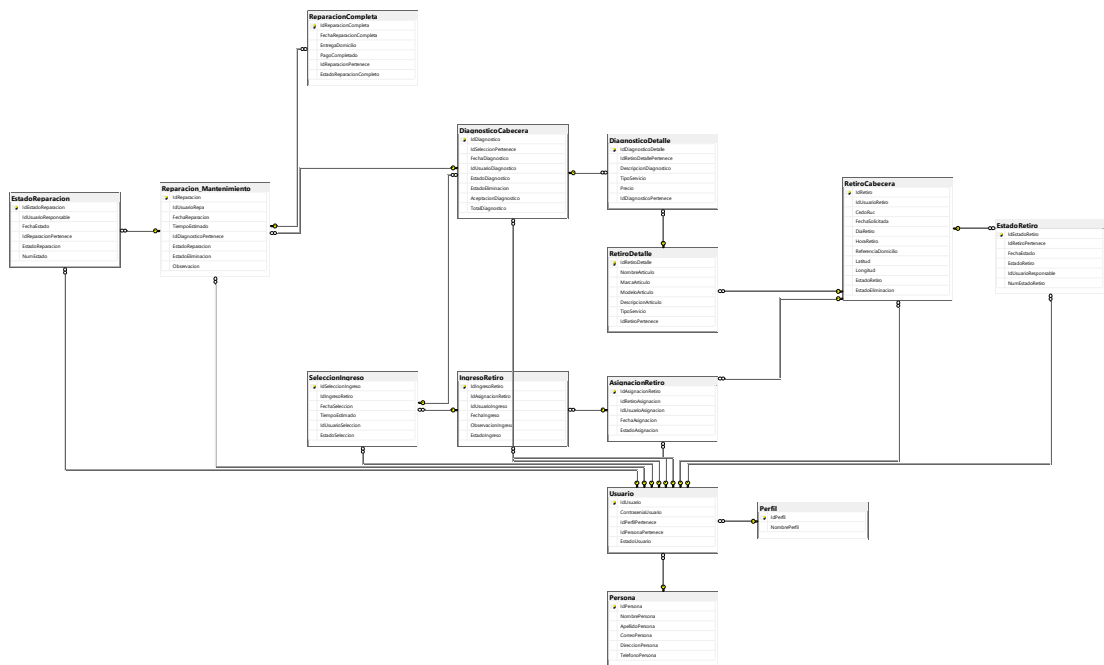
Checklist [Eliminar](#)

0%  Verificar el tipo de dato en las diferentes entidades  
 Verificar las relaciones

**Figura 28.** Pruebas del Diseño de la Base de Datos  
Elaborado por: Valeria Quispe

## Evidencia

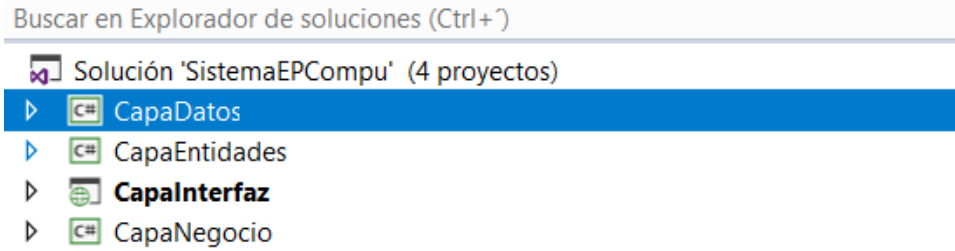
En la presente figura se presenta el modelo entidad relación del Sistema Web EPC-COMPU



**Figura 29.** Modelo Entidad-Relación de la Base de Datos  
Elaborado por: Valeria Quispe

## Codificación

En la solución realizada en Visual Studio 2017 se desarrolló un apartado con el fin de realizar una programación en capas, donde se puede identificar la capa de datos, entidades, negocio y presentación.



**Figura 30.** Arquitectura de Capas del Sistema  
**Elaborado por:** Valeria Quispe

A su vez la capa de datos genera la comunicación entre el back-end con el gestor de base de datos SQL SERVER, como se presenta en la cadena de conexión.

	Nombre	Tipo	Ámbito	Valor
	Conexion	(Cadena de ...	Aplicación	Data Source=.;Initial Catalog=SistemaEPC;Integrated Security=True
*				

**Figura 31.** Conexión a la Base de Datos  
**Elaborado por:** Valeria Quispe

Con ello una vez generada la conexión se puede acceder a la información almacenada en el gestor de base de datos mediante las sentencias SQL, eso se lo puede realizar desde cualquier clase de la capa de datos como:

```
SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
SqlCommand cmd = new SqlCommand();
cmd.Connection = conexion;
conexion.Open();
```

**Figura 32.** Acceso a la Base de Datos  
**Elaborado por:** Valeria Quispe

**Tarea Completada**



**Figura 33.** Diseño de la Base de Datos Completada  
**Elaborado por:** Valeria Quispe



**Figura 34.** Prueba del Diseño de la Base de Datos Completada  
**Elaborado por:** Valeria Quispe

## **HU02 Diseño de la página principal de presentación**

El diseño de la página principal se encuentra enfocado en la presentación de la información de la empresa EPC-Compu, detallando así los servicios, galería de fotos,

blog de noticias e información de la empresa, de esta forma se busca captar la atención del cliente adentro a los beneficios que puede obtener mediante el uso del sistema web.



**Figura 35.** Tareas y Subtareas del Diseño de la página principal de presentación  
**Elaborado por:** Valeria Quispe

## Prueba

La finalidad de las pruebas se encuentra enfocado en verificar si la información presentada en la página principal es de beneficio al usuario final, de la misma forma se revisa la correcta navegación en los diferentes apartados presentados como: Nosotros, Servicios, Ingresar, Galería e Inicio y sin dejar un lado lo más importante ver que el diseño sea de agrado para el cliente en este caso el representante de la empresa.



**Figura 36.** Pruebas del Diseño de la página principal de presentación  
**Elaborado por:** Valeria Quispe

### Evidencia

La página de inicio compuesta por el apartado de diferentes secciones permite al usuario conocer los servicios que dispone la empresa, visualizar una galería de fotos, conocer datos de contactos e información del equipo que compone la empresa. Incluso permite la navegación entre los formularios de creación de una nueva cuenta o simplemente el ingreso al sistema.

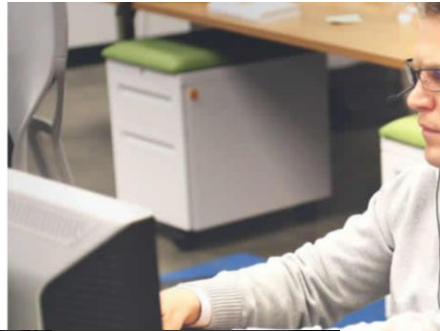


**Figura 37.** Página principal de Sistema EPC-Compu  
**Elaborado por:** Valeria Quispe

## Nosotros

EPC-COMPU es una empresa enfocada a brindar el servicio de mantenimiento y reparación de dispositivos electrónicos como: Impresoras, Portátiles, CPU, Tablets y Monitores dentro de la ciudad de Ambato.

Para mejora del servicio ofrecido, dispone a sus clientes la entrega y recepción de su dispositivo electrónico de puerta a puerta mediante la gestión georreferenciada, así como la gestión de la gestión de mantenimiento y reparación se la emplea mediante el sitio web ECP-COMPU.

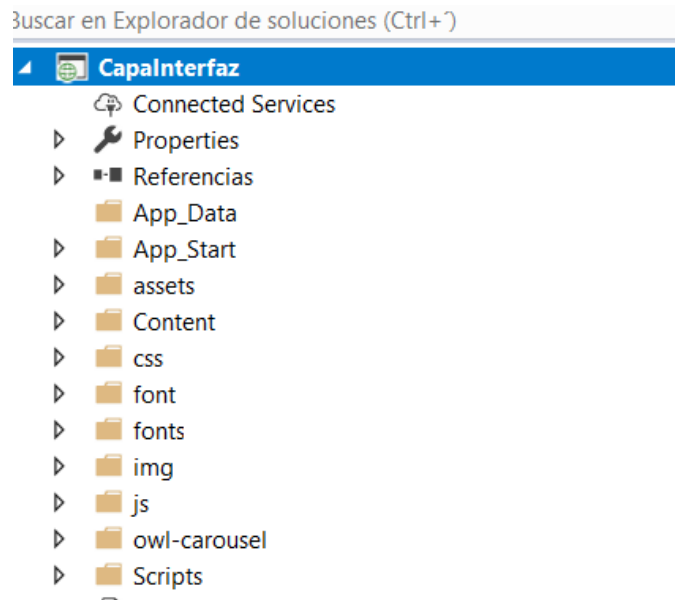


**Figura 38.** Secciones de la página principal del Sistema web  
**Elaborado por:** Valeria Quispe

## Codificación

Para la realización del diseño de la página principal, primero se define la plantilla Bootstrap que se será de ayuda en la presentación de información, agilidad en el desarrollo y adaptabilidad de los controles. Esto se lo puede descargar del sitio oficial o se puede indagar en diversas páginas con el fin de escoger la mejor opción. Con ello en el proyecto realizado en el entorno de desarrollo de Visual Studio se integra los estilos y los recursos que emplea la plantilla, mismo se lo ubicará en la capa de interfaz que se enfocará netamente en el front-end.





**Figura 39.** Recursos Bootstrap en la capa de interfaz  
**Elaborado por:** Valeria Quispe

Conforme se realice la integración, el formulario debe realizar referencias a los diferentes recursos, tanto así a recursos propios del proyecto o que se encuentre en la internet.

```

<!-- ===== Bootstrap Core CSS ===== -->
<link rel="stylesheet" href="css/bootstrap.min.css" type="text/css"/>
<!-- ===== fonts awesome ===== -->
<link rel="stylesheet" href="font/css/font-awesome.min.css" type="text/css"/>
<!-- ===== Plugin CSS ===== -->
<link rel="stylesheet" href="css/animate.min.css" type="text/css"/>
<!-- ===== Custom CSS ===== -->
<link rel="stylesheet" href="css/style.css" type="text/css"/>
<!-- ===== Owl Carousel Assets ===== -->
<link href="owl-carousel/owl.carousel.css" rel="stylesheet"/>
<link href="owl-carousel/owl.theme.css" rel="stylesheet"/>
<link rel="stylesheet" href="css/isotope-docs.css" media="screen"/>
<link rel="stylesheet" href="css/baguetteBox.css"/>

```

**Figura 40.** Referencias de los recursos de Bootstrap  
**Elaborado por:** Valeria Quispe

La navegación entre las diferentes secciones, se lo realiza mediante la redirección href mismo que es utilizado por los recursos ofrecidos por Bootstrap.

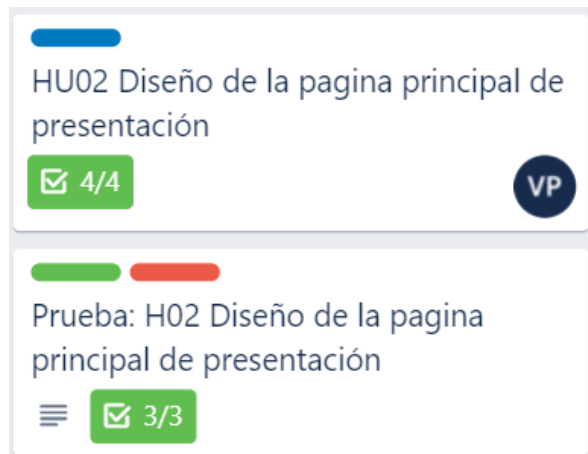
```

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav navbar-right">
    <li>
      <a class="page-scroll" href="#home">Inicio</a>
    </li>
    <li>
      <a class="page-scroll" href="#about">Nosotros</a>
    </li>
    <li>
      <a class="page-scroll" href="#Services">Servicios</a>
    </li>
    <li>
      <a class="page-scroll" href="#Gallery">Galería</a>
    </li>
    <li>
      <a class="page-scroll" href="#blog">Blog</a>
    </li>
    <li>
      <a class="page-scroll" href="#contact">Ingresar</a>
    </li>
  </ul>
</div>

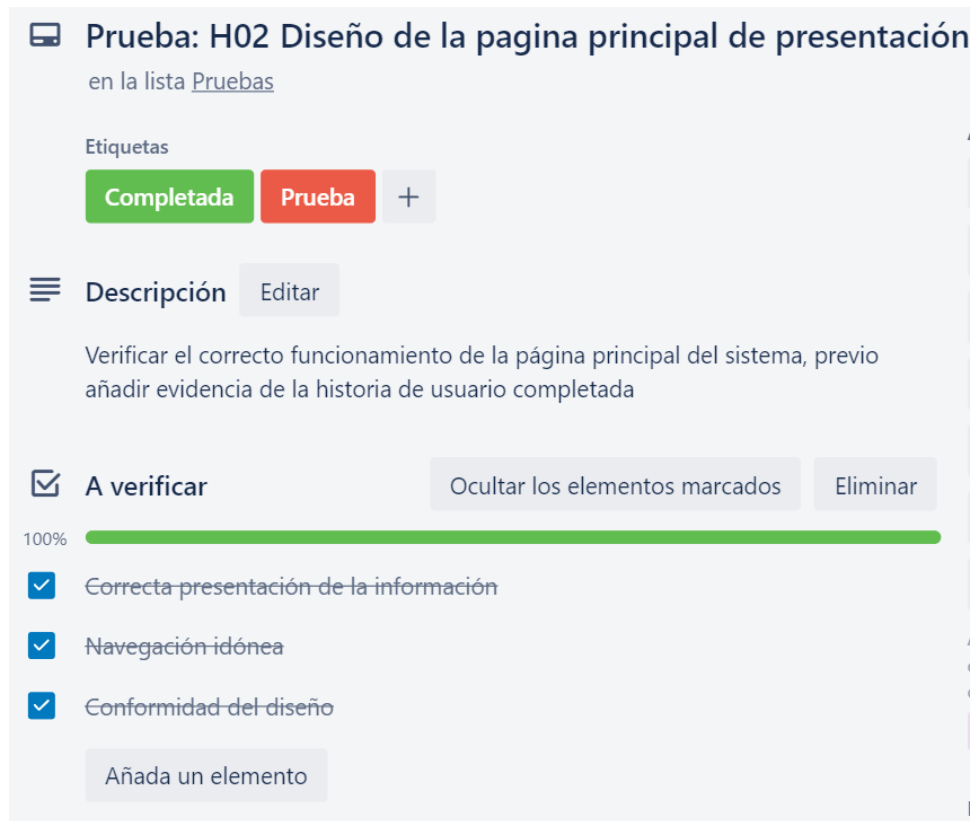
```

**Figura 41.** Navegación de secciones, página principal  
**Elaborado por:** Valeria Quispe

## Tarea Completa



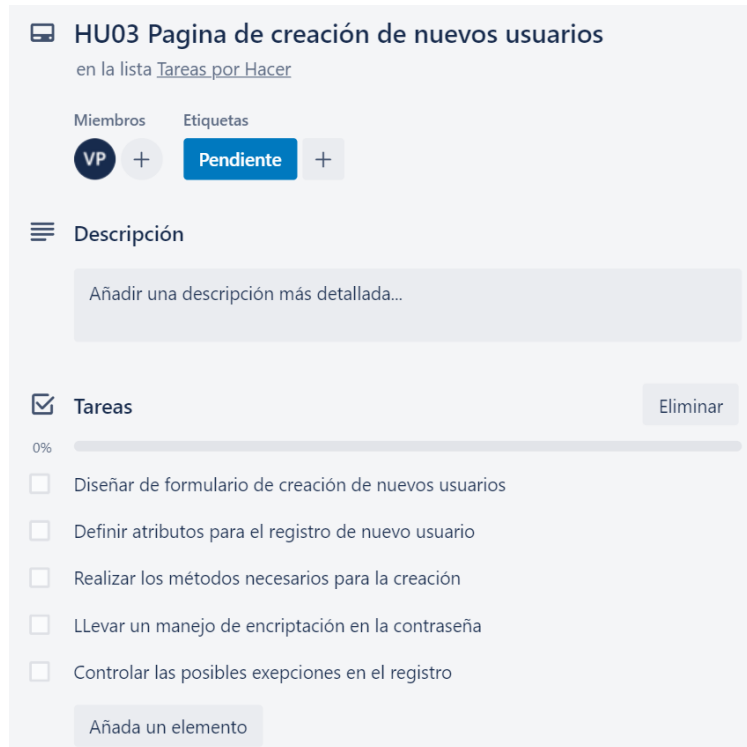
**Figura 42.** Diseño de la página principal de presentación Completada  
**Elaborado por:** Valeria Quispe



**Figura 43.** Pruebas del Diseño de la Página Principal de presentación  
**Elaborado por:** Valeria Quispe

### **H03 Página de creación de nuevos usuarios**

La página de creación de nuevos usuarios se encuentra enfocado en la presentación de un formulario, con la finalidad de receptar la información necesaria para realizar el registro de un nuevo cliente, con esto el usuario podrá acceder a todos los servicios que ofrece el Sistema Web EPC-COMPU.



**Figura 44.** Página de creación de nuevos usuarios  
**Elaborado por:** Valeria Quispe

## Prueba

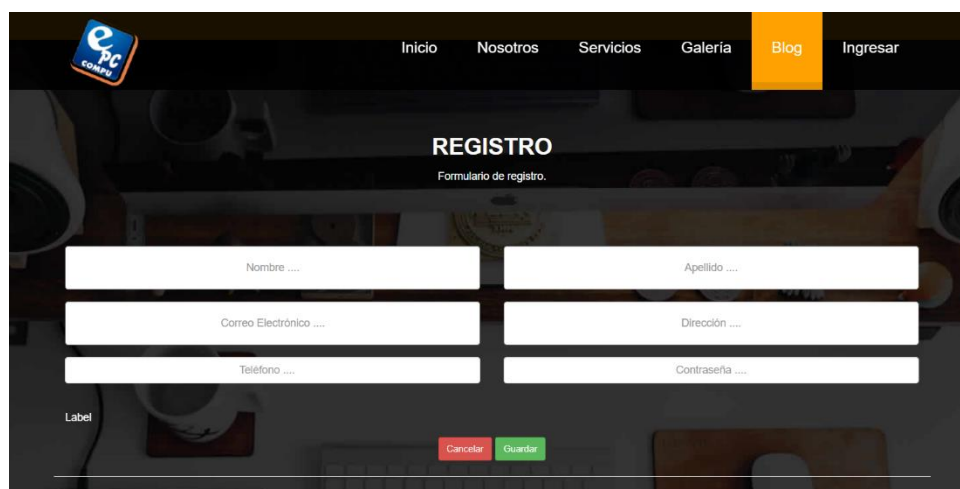
La finalidad de las pruebas concierne a verificar la funcionalidad del registro de un nuevo usuario, esto debido a que ingresará la información del nuevo cliente en los campos presentados y una vez finalizado se procederá con el registro. Todo a favor de que el nuevo usuario pueda acceder a los servicios del sistema mediante el usuario creado.



**Figura 45.** Pruebas de la Página de creación de nuevos usuarios  
**Elaborado por:** Valeria Quispe

### Evidencia

El formulario de creación de nuevo usuario presenta un apartado para el ingreso de la información personal del usuario, como: nombre, apellido, correo electrónico, dirección, teléfono y contraseña con ello el usuario realizará el previo registro y si es caso de ser correcta toda la información se realizará el proceso de guardado en el servidor de base de datos o caso contrario se presentará un mensaje de aviso con la finalidad de que el usuario pueda completar el registro de manera correcta.



**Figura 46.** Formulario de registro de nuevo usuario

**Elaborado por:** Valeria Quispe

## Codificación

Conforme a cumplir el proceso de registro se desarrolló el método de encriptación de la contraseña para mantener la seguridad de los datos esenciales del usuario donde se detalla a continuación el método, para lo cual se empleó una llave que será utilizada para su previa desencriptación.

```
public string Encrypt(string plainText)
{
    if (plainText == null)
    {
        return null;
    }
    // Get the bytes of the string
    var bytesToBeEncrypted = Encoding.UTF8.GetBytes(plainText);
    var passwordBytes = Encoding.UTF8.GetBytes(password);

    // Hash the password with SHA256
    passwordBytes = SHA512.Create().ComputeHash(passwordBytes);

    var bytesEncrypted = Encrypt(bytesToBeEncrypted, passwordBytes);

    return Convert.ToBase64String(bytesEncrypted);
}
```

**Figura 47.** Método de encriptación  
**Elaborado por:** Valeria Quispe

```
public byte[] Encrypt(byte[] bytesToBeEncrypted, byte[] passwordBytes)
{
    byte[] encryptedBytes = null;
    var saltBytes = new byte[] { 1, 2, 3, 4, 5, 6, 7, 8 };

    using (MemoryStream ms = new MemoryStream())
    {
        using (RijndaelManaged AES = new RijndaelManaged())
        {
            var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);
            AES.KeySize = 256;
            AES.BlockSize = 128;
            AES.Key = key.GetBytes(AES.KeySize / 8);
            AES.IV = key.GetBytes(AES.BlockSize / 8);
            AES.Mode = CipherMode.CBC;

            using (var cs = new CryptoStream(ms, AES.CreateEncryptor(), CryptoStreamMode.Write))
            {
                cs.Write(bytesToBeEncrypted, 0, bytesToBeEncrypted.Length);
                cs.Close();
            }
            encryptedBytes = ms.ToArray();
        }
    }
    return encryptedBytes;
}
```

**Figura 48.** Algoritmo de encriptación  
**Elaborado por:** Valeria Quispe

Consecuentemente se lleva a cabo la recolección de la información del usuario en la capa de presentación conforme se presenta en el método registrarNuevoUsuario().

```
protected void registrarNuevoUsuario()
{
    //Registrar Persona
    EPersona personaNueva = new EPersona();
    personaNueva.NombrePersona = TextBoxNombrePersona.Text;
    personaNueva.ApellidoPersona = TextBoxApellidoPersona.Text;
    personaNueva.CorreoPersona = TextBoxCorreoPersona.Text;
    personaNueva.TelefonoPersona = TextBoxTelefonoPersona.Text;
    personaNueva.DireccionPersona = TextBoxDireccionPersona.Text;
    personaNueva = NPersona.GuardarPersonaNegocio(personaNueva);
    if (personaNueva == null)
    {
        LabelMensaje.Text = "Registro no completado";
    }
    //Registrar Usuario
    EPerfil perfil = new EPerfil();
    EUsuario nuevo = new EUsuario();
    nuevo.IdPerfilPertenece = 2;
    nuevo.ContraseniaUsuario = metodos.Encrypt(TextBoxContrasenia.Text);
    nuevo.IdPersonaPertenece = personaNueva.IdPersona;
    nuevo.EstadoUsuario = 1;
    nuevo = NUsuario.GuardarUsuarioNegocio(nuevo);
    if (nuevo == null)
    {
        LabelMensaje.Text = "Registro no completado";
    }
    else
    {
        LabelMensaje.Text = "Registro completado";
        MultiView1.ActiveViewIndex = 0;
    }
}
```

**Figura 49.** Método registra nuevo usuario  
**Elaborado por:** Valeria Quispe

El método de registrar nuevo usuario recupera los datos de las componentes web, con eso hace una creación de entidades las cuales son procesadas y enviadas a la capa de negocio, esto a su vez son procesadas en dicha para llegar así a la capa de datos que es donde se realiza el registro en el servidor de base datos, para ello se presenta los métodos de registro de la capa de datos.

```

public static EPersona GuardarPersonaDatos(EPersona personaNueva)
{
    try
    {
        SqlConnection connection = new SqlConnection(Settings1.Default.Conexion);
        connection.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = connection;
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = @"INSERT INTO [dbo].[Persona]
([NombrePersona]
,[ApellidoPersona]
,[CorreoPersona]
,[DireccionPersona]
,[TelefonoPersona])
VALUES
(@NombrePersona,@ApellidoPersona,@CorreoPersona,@DireccionPersona,@TelefonoPersona);SELECT SCOPE_IDENTITY()";
        cmd.Parameters.AddWithValue("@NombrePersona", personaNueva.NombrePersona);
        cmd.Parameters.AddWithValue("@ApellidoPersona", personaNueva.ApellidoPersona);
        cmd.Parameters.AddWithValue("@CorreoPersona", personaNueva.CorreoPersona);
        cmd.Parameters.AddWithValue("@DireccionPersona", personaNueva.DireccionPersona);
        cmd.Parameters.AddWithValue("@TelefonoPersona", personaNueva.TelefonoPersona);

        var idpersona = Convert.ToInt32(cmd.ExecuteScalar());
        personaNueva.IdPersona = idpersona;
        connection.Close();
        return personaNueva;
    }
    catch (Exception)
    {
        return null;
    }
}

```

**Figura 50.** Método de guardar persona en la capa de datos  
**Elaborado por:** Valeria Quispe

```

public static EUsuario GuardarUsuarioDatos(EUsuario nuevo)
{
    try
    {
        SqlConnection connection = new System.Data.SqlClient.SqlConnection(Settings1.Default.Conexion);
        connection.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = connection;
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = @"INSERT INTO [dbo].[Usuario]
([ContraseniaUsuario]
,[IdPerfilPertenece]
,[IdPersonaPertenece]
,[EstadoUsuario])
VALUES(@Contrasenia,@Perfil,@Persona,@Estado);
SELECT SCOPE_IDENTITY()";
        cmd.Parameters.AddWithValue("@Contrasenia", nuevo.ContraseniaUsuario);
        cmd.Parameters.AddWithValue("@Perfil", nuevo.IdPerfilPertenece);
        cmd.Parameters.AddWithValue("@Persona", nuevo.IdPersonaPertenece);
        cmd.Parameters.AddWithValue("@Estado", nuevo.EstadoUsuario);
        var idusuario = Convert.ToInt32(cmd.ExecuteScalar());
        nuevo.IdUsuario = idusuario;

        return nuevo;
    }
    catch (Exception ex)
    {
        return null;
    }
}

```

**Figura 51.** Método de guardar usuario en la capa de datos  
**Elaborado por:** Valeria Quispe

Dichos métodos realizan el registro del nuevo usuario, pero en caso de ser positivo la operación el cliente podrá observar el formulario de acceso al sistema y de ser

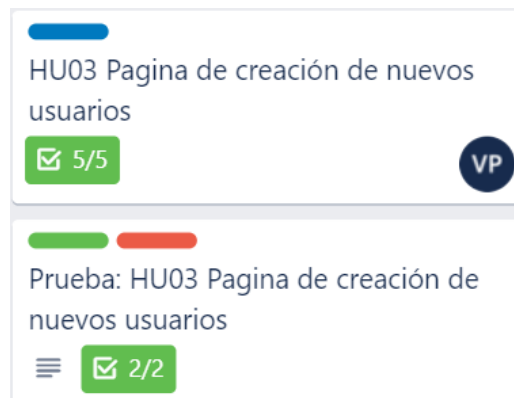


erróneo se presentará un mensaje, de la misma forma para evitar cualquier problema y que no se detenga el funcionamiento se realiza el control de cualquier excepción que pudiese presentar con el siguiente control.

```
protected void ButtonGuardarUsuario_Click(object sender, EventArgs e)
{
    try
    {
        registrarNuevoUsuario();
    }
    catch (Exception)
    {
        LabelMensaje.Text = "¡Ingreso Erróneo, intente más tarde!";
    }
}
```

**Figura 52.** Control de excepción registro de nuevo usuario  
**Elaborado por:** Valeria Quispe

### Tarea Completa



**Figura 53.** Página de creación de nuevos usuarios completada  
**Elaborado por:** Valeria Quispe



**Figura 54.** Pruebas de la página de creación de nuevos usuarios completada

**Elaborado por:** Valeria Quispe

#### **H04 Acceso al sistema**

El formulario de acceso al sistema centra en presentar al usuario un diseño de fácil comprensión, donde el usuario podrá acceder a los servicios, funciones que ofrece la empresa mediante el sistema web esto a partir de realizar el logeo pertinente y de validar las credenciales de acceso como lo son correo electrónico y contraseña.

The screenshot shows a task management interface for a task titled "HU04 Acceso al sistema". The task is located in the "Tareas por Hacer" list. It has one member, "VP", and is labeled "Pendiente". The description field is empty, with a placeholder text "Añadir una descripción más detallada...". The task is currently 0% complete. The task list includes four items: "Definir diseño de acceso", "Definir campos de validación", "Creación de métodos para controlar el acceso", and "Controlar las excepciones". There is an "Añadir un elemento" button at the bottom of the task list.

**HU04 Acceso al sistema**  
en la lista [Tareas por Hacer](#)

Miembros: **VP** + Etiquetas: **Pendiente** +

Descripción

Añadir una descripción más detallada...

**Tareas** Eliminar

0%

- Definir diseño de acceso
- Definir campos de validación
- Creación de métodos para controlar el acceso
- Controlar las excepciones

Añada un elemento

**Figura 55. Acceso al Sistema**  
**Elaborado por:** Valeria Quispe

## Prueba

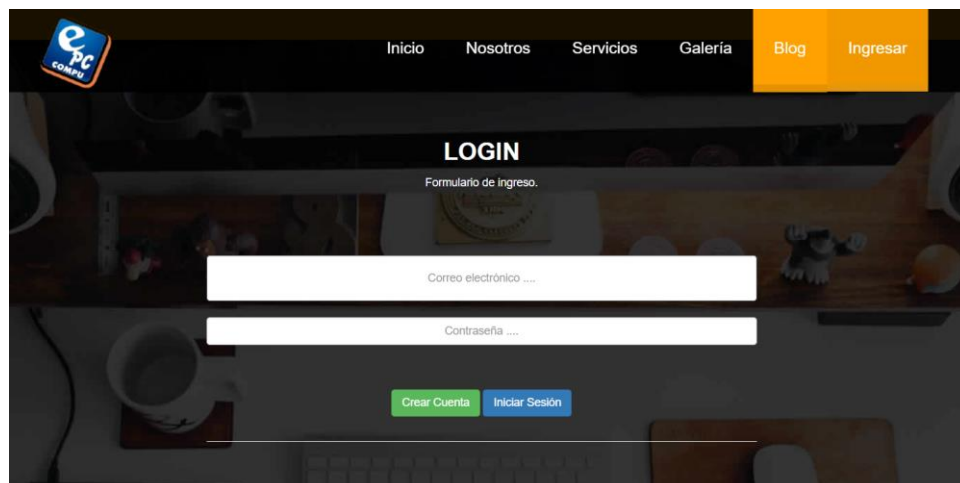
La finalidad de las pruebas en el formulario de acceso se enfoca en realizar la correcta validación de las credenciales del usuario a ingresar al sistema, esto debido a que pasará por un proceso de codificación y de decodificación de la contraseña como norma de seguridad en el cifrado de credenciales sensibles.



**Figura 56.** Pruebas de Acceso al sistema  
**Elaborado por:** Valeria Quispe

### Evidencia

El formulario de acceso al sistema presenta el diseño para recibir la información de correo electrónico y contraseña del usuario para poder realizar la validación de credenciales.



**Figura 57.** Formulario de acceso al sistema  
**Elaborado por:** Valeria Quispe

## Codificación

Conforme a realizar el desarrollo de los métodos que ayuden a validar el ingreso del usuario, se generó la parte de descryptación esto para mantener la integridad de los datos sensible tal como se muestra a continuación:

```
public string Decrypt(string encryptedText)
{
    if (encryptedText == null)
    {
        return null;
    }
    // Get the bytes of the string
    var bytesToBeDecrypted = Convert.FromBase64String(encryptedText);
    var passwordBytes = Encoding.UTF8.GetBytes(password);

    passwordBytes = SHA512.Create().ComputeHash(passwordBytes);

    var bytesDecrypted = Decrypt(bytesToBeDecrypted, passwordBytes);

    return Encoding.UTF8.GetString(bytesDecrypted);
}
```

**Figura 58.** Método de descryptación  
Elaborado por: Valeria Quispe

```
private static byte[] Decrypt(byte[] bytesToBeDecrypted, byte[] passwordBytes)
{
    byte[] decryptedBytes = null;
    var saltBytes = new byte[] { 1, 2, 3, 4, 5, 6, 7, 8 };
    try
    {
        using (MemoryStream ms = new MemoryStream())
        {
            using (RijndaelManaged AES = new RijndaelManaged())
            {
                var key = new Rfc2898DeriveBytes(passwordBytes, saltBytes, 1000);
                AES.KeySize = 256;
                AES.BlockSize = 128;
                AES.Key = key.GetBytes(AES.KeySize / 8);
                AES.IV = key.GetBytes(AES.BlockSize / 8);
                AES.Mode = CipherMode.CBC;

                using (var cs = new CryptoStream(ms, AES.CreateDecryptor(), CryptoStreamMode.Write))
                {
                    cs.Write(bytesToBeDecrypted, 0, bytesToBeDecrypted.Length);
                    cs.Close();
                }
                decryptedBytes = ms.ToArray();
            }
        }
    }
    catch (Exception)
    {
    }
    return decryptedBytes;
}
```

**Figura 59.** Algoritmo de descryptación  
Elaborado por: Valeria Quispe

De esta manera se desarrolla el método ValidarIngresoUsuario() en la capa de interfaz, que es responsable de recibir los datos como correo electrónico y contraseña para realizar la validación pertinente, de esta forma se hace el llamado al método DevolverUsuarioIdPersona() para verificar los datos obtenidos del servidor de base datos con las credenciales ingresadas y en caso de ser positiva la operación se realizará la redirección a la página de inicio y de ser erróneo el ingreso se presentara un mensaje.

```
private void ValidarIngresoUsuario()
{
    try
    {
        personaIngresa = NPersona.DevolverPersonaxCorreoNegocio(TextBoxCorreoIngreso.Text);
        Session["Persona"] = personaIngresa;
        usuarioIngresa = IUusuario.DevolverUsuarioIdPersona(personaIngresa.IdPersona);
        Session["Usuario"] = usuarioIngresa;

        if (personaIngresa.CorreoPersona == TextBoxCorreoIngreso.Text && metodos.Decrypt(usuarioIngresa.ContraseniaUsuario) == TextBoxContraseniaIngreso.Text)
        {
            Response.Redirect("IPaginaEPC.aspx", true);
            limpiarCampos();
        }
        else
        {
            LabelMensajeIngreso.Text = ";Credenciales Incorrectas!";
            limpiarCampos();
        }
    }
    catch (Exception)
    {
        LabelMensaje.Text = ";Ingreso Erróneo, intente más tarde!";
    }
}
}
```

**Figura 60.** Método de validar ingreso en la capa de interfaz  
**Elaborado por:** Valeria Quispe

```
public static EUsuario DevolverUsuarioIdPersona(int idPersona)
{
    return DUusuario.DevolverUsuarioIdPersonaDatos(idPersona);
}
```

**Figura 61.** Método de obtener usuario en la capa de negocio  
**Elaborado por:** Valeria Quispe

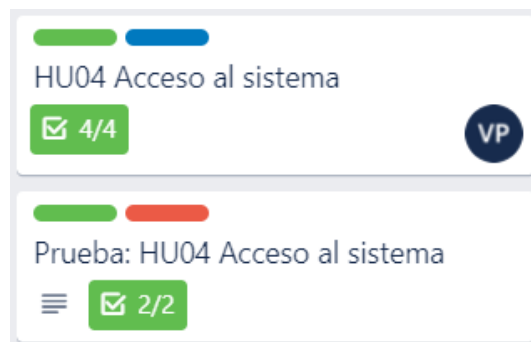
```

public static EUsuario DevolverUsuarioxIdPersonaDatos(int idPersona)
{
    try
    {
        EUsuario usuario = new EUsuario();
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdUsuario]
                            ,[ContraseniaUsuario]
                            ,[IdPerfilPertenece]
                            ,[IdPersonaPertenece]
                            FROM [dbo].[Usuario]
                            where IdPersonaPertenece=@id";
        cmd.Parameters.AddWithValue("@id", idPersona);
        cmd.CommandType = System.Data.CommandType.Text;
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                usuario.IdUsuario = Convert.ToInt32(dr["IdUsuario"].ToString());
                usuario.ContraseniaUsuario = dr["ContraseniaUsuario"].ToString();
                usuario.IdPerfilPertenece = Convert.ToInt32(dr["IdPerfilPertenece"].ToString());
                usuario.IdPersonaPertenece = Convert.ToInt32(dr["IdPersonaPertenece"].ToString());
            }
        }
        conexion.Close();
        return usuario;
    }catch (SqlException ex)
    {
        return null;
    }catch (Exception)
    {
        return null;
    }
}

```

**Figura 62.** Método de obtener usuario en la capa de datos  
**Elaborado por:** Valeria Quispe

### Tarea completada



**Figura 63.** Acceso al Sistema Completada  
**Elaborado por:** Valeria Quispe

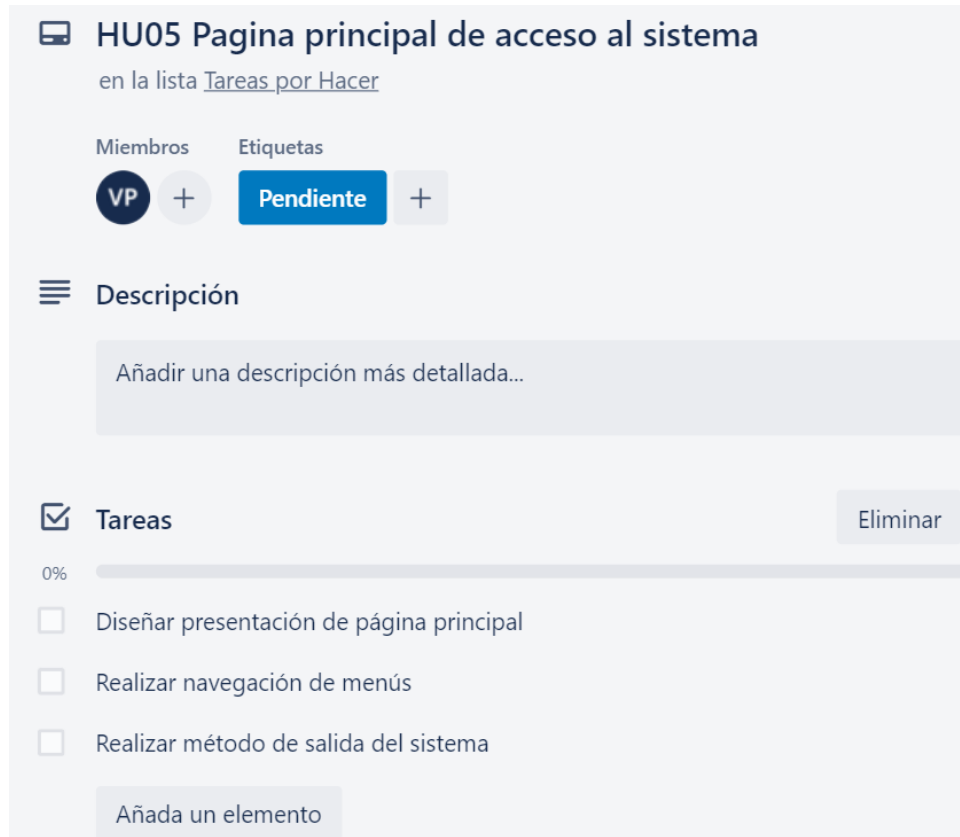


**Figura 64.** Pruebas de Acceso al Sistema completada  
**Elaborado por:** Valeria Quispe

### **H05 Página principal de acceso al sistema**

La página principal de acceso se encuentra enfocado en presentar información del sistema, así como el menú correspondiente al perfil del usuario que ingreso esto a favor de que pueda manejar las opciones necesarias para utilizar los servicios, de la misma forma presenta un apartado para poder visualizar el perfil de información personal y modificar los campos posibles.

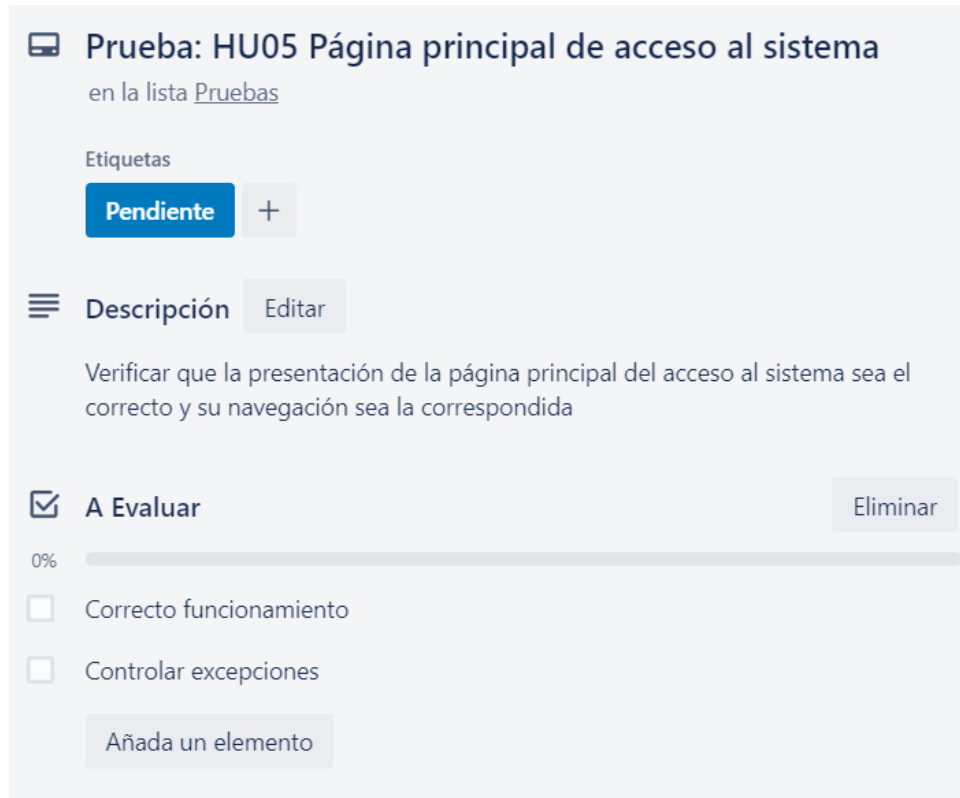




**Figura 65.** Página principal de acceso al sistema  
**Elaborado por:** Valeria Quispe

### Prueba

El fin de llevar a cabo las pruebas son de verificar la correcta habilitación de los menús conforme a los perfiles del usuario que ingrese y ver que la información presentada en la página de inicio sea de beneficio para el usuario.



**Figura 66.** Pruebas de la página principal de acceso al sistema  
**Elaborado por:** Valeria Quispe

## Evidencia

La página de inicio presenta un apartado de los menús, mismo se habilitarán conforme al perfil del usuario que realiza el ingreso al sistema, con esto a su vez se tendrá la posibilidad de poder visualizar la información personal como editar los campos habilitados



**Figura 67.** Página principal de acceso al sistema

**Elaborado por:** Valeria Quispe

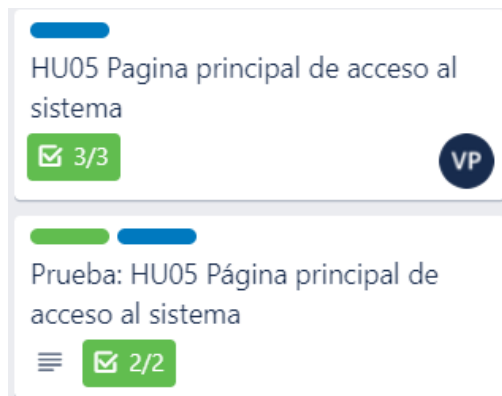
## Codificación

El desarrollo se lo realiza primero con el diseño de la página, definiendo así las secciones de los menús como se muestra en la figura

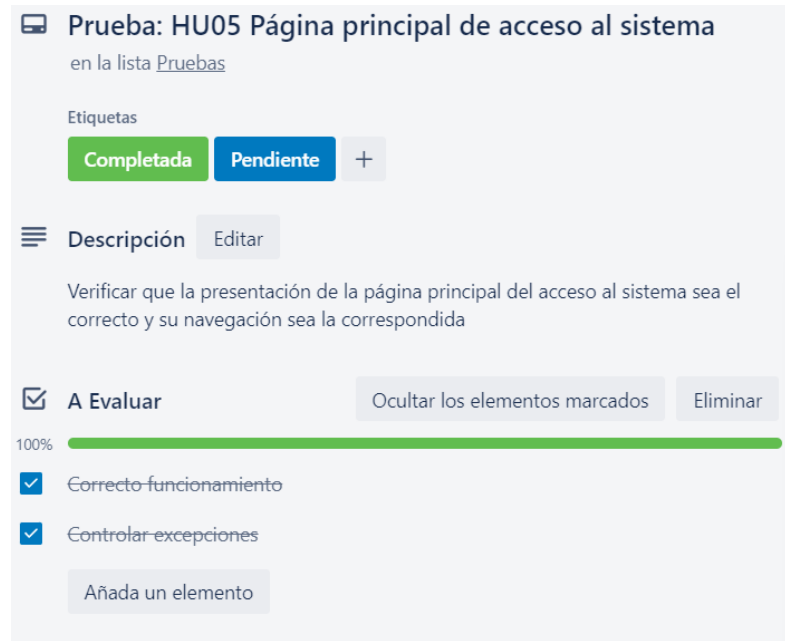


**Figura 68.** Diseño y secciones de menú de la página de inicio  
**Elaborado por:** Valeria Quispe

## Tarea Completada



**Figura 69.** Página principal de acceso al sistema completada  
**Elaborado por:** Valeria Quispe



**Figura 70.** Pruebas de la página principal de acceso al sistema completada

**Elaborado por:** Valeria Quispe

### **H06 Gestión de retiros**

La gestión de retiros se enfoca en la organización de las operaciones que puede realizar el usuario con respecto a los retiros como es el, solicitar, ver estado, asignar, ver asignación e ingresar.



**Figura 71. Gestión de Retiros**  
**Elaborado por:** Valeria Quispe

Conforme a obtener una mejor organización con respecto a la realización de la gestión de retiros, se subdivide en tareas para el desarrollo pertinente de cada operación. Con ello se inicia el desarrollo de la subtarea solicitar retiro que tendrá como finalidad presentar un formulario para que el usuario cliente pueda solicitar el retiro a domicilio de los dispositivos a revisar.

The screenshot shows a user interface for a sub-task titled "Sub-Tarea HU06: Solicitar Retiro". At the top, there is a header with a folder icon, the title, and a link to "Subtareas" with a circular icon. Below this is a "Descripción" section with a menu icon and an "Editar" button. The description text reads: "Realizar el formulario donde el usuario final o cliente pueda solicitar el retiro de los dispositivos, donde se detalle la información personal, de los dispositivos y su respectiva ubicación".

The "Tareas" section features a checkmark icon, the title "Tareas", and an "Eliminar" button. A progress bar shows "0%". Below the progress bar is a list of five tasks, each with an unchecked checkbox:

- Diseño de formulario
- Definir método de obtención de coordenadas geográficas
- Definir métodos funcionamiento, agregar dispositivo nuevo
- Realizar método de registro
- Controlar excepciones

At the bottom of the task list, there is a note: "Hay modificaciones no guardadas en este campo. [Visualizar ediciones](#) - [Descartar](#)". Below the note is a button labeled "Añada un elemento".

**Figura 72.** Subtareas de la historia de usuario HU06 Solicitar Retiro  
**Elaborado por:** Valeria Quispe

La subtarea estado de retiros se enfocará en presentar el formulario donde el cliente podrá visualizar de manera detalla el estado de los retiros que aún no se encuentran finalizados.



**Figura 73.** Subtarea de la historia de usuario HU06 - Estado de Retiros

**Elaborado por:** Valeria Quispe

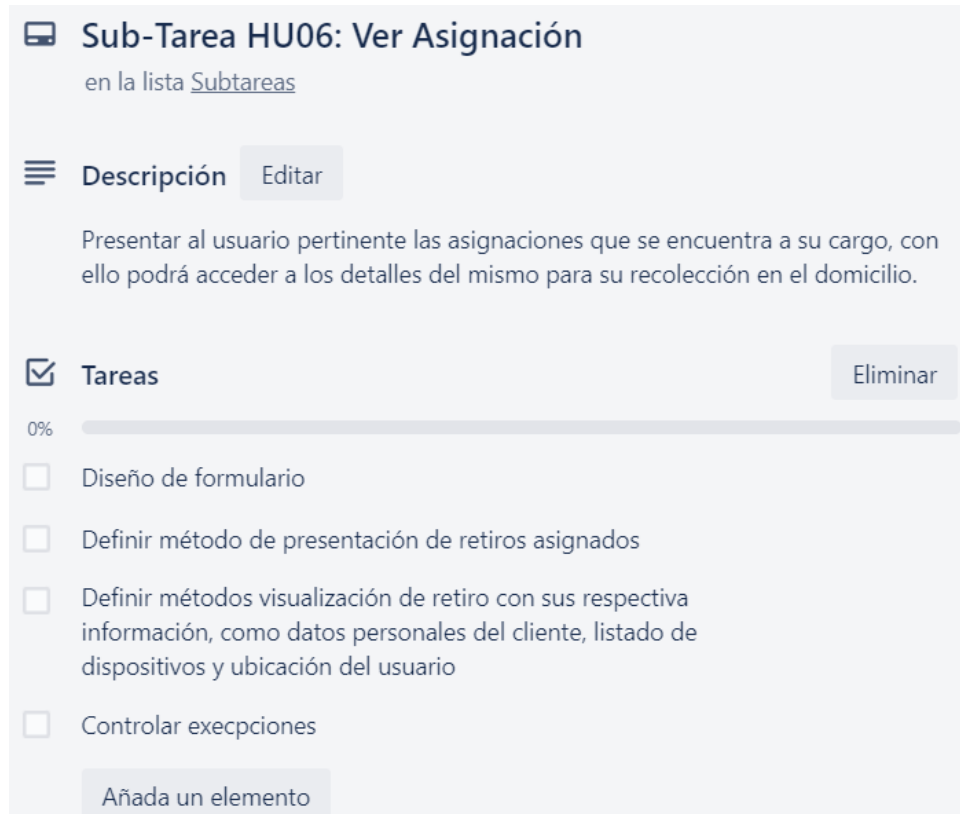
La subtarea asignar retiro se enfocará en el desarrollo de la operación en el que personal pertinente podrá asignar un retiro solicitado por el cliente al personal de entrega y recepción con el fin de recolectar los dispositivos tecnológicos para su revisión.



**Figura 74.** Sub-Tareas de la Historia de Usuario HU06 - Asignar Retiro  
**Elaborado por:** Valeria Quispe

La subtarea ver asignación se encuentra enfocado en presentar la operación poder visualizar una asignación pendiente al usuario que fuere asignado y con ello podrá tener los datos del cliente, domicilio, dispositivos que requieren una revisión y lo más importante tendrá una vista minuciosa de la ubicación del domicilio para trasladarse al retiro.





**Figura 75.** Sub-Tarea de la Historia de Usuario HU06 - Ver Asignación  
**Elaborado por:** Valeria Quispe

La subtarea ingresar retiro se enfoca en presentar la operación donde el usuario pertinente realiza el registro de la llegada de los dispositivos de forma correcta o de ser el caso poner cualquier observación que pueda presentarse.

The screenshot shows a user interface for a task titled "Sub-Tarea HU06: Ingresar Retiro". At the top, there is a header with a folder icon and the text "Sub-Tarea HU06: Ingresar Retiro" and "en la lista [Subtareas](#)". Below this is a section for "Descripción" with a menu icon and an "Editar" button. The description text reads: "Presentar al usuario los ingresos realizados, así como el formulario para los nuevos ingresos de los retiros en la empresa." Below the description is a "Tareas" section with a checkmark icon and an "Eliminar" button. A progress bar shows "0%". The task list includes: "Diseño de formulario", "Definir método de presentación de ingresos", "Definir métodos funcionamiento, retiros pendientes, personal de recepción y entrega", "Realizar método de registro", and "Controlar excepciones". A note below the third task says: "Hay modificaciones no guardadas en este campo. [Visualizar ediciones](#) - [Descartar](#)". At the bottom is a button labeled "Añada un elemento".

**Figura 76.** Sub-Tarea de la Historia de Usuario HU06 - Ingresar Retiro

**Elaborado por:** Valeria Quispe

### **Prueba**

La finalidad de las pruebas es verificar que las operaciones pertenecientes a la gestión de retiros cumplan con su finalidad de manera correcta, estos a su vez cumplir con el servicio puesto a disposición por la empresa EPC-COMPU.



**Figura 77.** Pruebas de la Gestión de Retiros  
**Elaborado por:** Valeria Quispe

## **Evidencia**

### **Subtarea solicitar retiro**

El formulario de solicitar retiro se encuentra enfocado a que el cliente pueda registrar el retiro a domicilio de los dispositivos tecnológicos a evaluar, donde detallará información personal, ubicación de su hogar, entre otros detalles que ayuden a coordinar el retiro, además ingresará un listado de los dispositivos donde ingresará un detalle como marca, modelo, tipo de trabajo y descripción del artículo.

**Solicitar Retiro en Domicilio**

---

**Información del Retiro**

Nombre:	<input type="text" value="Valeria Quispe"/>	Correo:	<input type="text" value="marco"/>	Ubicación de Domicilio
Teléfono:	<input type="text" value="0979398197"/>	Usuario:	<input type="text" value="3"/>	Latitud: <input type="text"/>
Cédula u RUC:	<input type="text"/>	Fecha:	<input type="text" value="22/06/2022 23:55"/>	Longitud: <input type="text"/>
Día Retiro:	<input type="text" value="Seleccionar"/>	Hora Retiro:	<input type="text" value="Seleccionar"/>	
Referencia Domicilio:	<input type="text"/>			

Artículos de Retiro

---

**Información del Artículo**

Nombre:	<input type="text"/>	Marca:	<input type="text"/>	Lista de Artículos
Modelo:	<input type="text"/>	Tipo Trabajo:	<input type="text" value="Desconocido"/>	<input type="button" value="Generar Retiro"/>
Descripción:	<input type="text"/>			

LA LISTA DE ARTÍCULOS ESTÁ VACÍA

**Figura 78.** Formulario de solicitar retiro  
**Elaborado por:** Valeria Quispe

### Codificación

En lo principal se destaca el uso del Api de Google Maps, utilizado en la recepción de la ubicación del cliente, para el cual se genera una llave empleada en la página master, con esto se podrá acceder a las funciones que ofrece dicho api.

```
<script type="text/javascript"
src="https://maps.google.com/maps/api/js?sensor=false&libraries=places&key=AIzaSyDwfhNnNbH1-Oi5Pv7T9AVPICu8RW1AqwM&libraries=visualization"></script>
```

**Figura 79.** Uso de la Api de Google Maps  
**Elaborado por:** Valeria Quispe

### Llave de la Api de Google Maps

Con ello se define el método desarrollado en JavaScript para la presentación del mapa dinámico, donde el cliente podrá movilizarse en el mapa y generar las coordenadas geográficas obteniendo así latitud y longitud del domicilio del interesado.

```

<script>
var vMarker
var map
map = new google.maps.Map(document.getElementById('map_canvas'), {
    zoom: 14,
    center: new google.maps.LatLng(-1.268921, -78.623623),
    mapTypeId: google.maps.MapTypeId.ROADMAP
});
vMarker = new google.maps.Marker({
    position: new google.maps.LatLng(-1.268921, -78.623623),
    draggable: true
});
google.maps.event.addListener(vMarker, 'dragend', function (evt) {
    $("#<%=TextBoxLatitud.ClientID%>").val(evt.latLng.lat().toFixed(6));
    $("#<%=TextBoxLongitud.ClientID%>").val(evt.latLng.lng().toFixed(6));
    map.panTo(evt.latLng);
});
map.setCenter(vMarker.position);
vMarker.setMap(map);
$("#txtCiudad, #txtEstado, #txtDireccion").change(function () {
    movePin();
});
function movePin() {
    var geocoder = new google.maps.Geocoder();
    var textSelectM = $("#txtCiudad").text();
    var textSelectE = $("#txtEstado").val();
    var inputAddress = $("#txtDireccion").val() + ' ' + textSelectM + ' ' + textSelectE;
    geocoder.geocode({
        "address": inputAddress
    }, function (results, status) {
        if (status == google.maps.GeocoderStatus.OK) {
            vMarker.setPosition(new google.maps.LatLng(results[0].geometry.location.lat(), results[0].geometry.location.lng()));
            map.panTo(new google.maps.LatLng(results[0].geometry.location.lat(), results[0].geometry.location.lng()));
            $("#<%=TextBoxLatitud.ClientID%>").val(results[0].geometry.location.lat());
            $("#<%=TextBoxLongitud.ClientID%>").val(results[0].geometry.location.lng());
        }
    });
}
});

```

**Figura 80. Llave de la Api de Google Maps**  
**Elaborado por: Valeria Quispe**

Consecuentemente se lleva a cabo el método para cumplir con la funcionalidad de agregar un artículo al listado de dispositivos a revisar, esto para luego seguir con el registro de la solicitud. El método denominado como AgregarArticulo() se divide en dos secciones la primera en caso de que no se pueda agregar de forma correcta y otra al ser agregado de forma correcta.

```

private void AgregarArticulo()
{
    try
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["dt"];
        DataRow[] dr;
        string nombre = "", marca = "", modelo = "", descripcion = "", tipo = "";
        if (TextBoxNombreArticulo.Text == "" || TextBoxModeloArticulo.Text == "" ||
            TextBoxMarcaArticulo.Text == "" || TextBoxDescripcionArticulo.Text == ""
            || DropDownListTipoTrabajo.SelectedIndex == 0)
        {
            LabelMensaje.ForeColor = System.Drawing.Color.Red;
            LabelMensaje.Text = "<b>Mensaje</b> : Datos incompletos para agregar articulo";
            ECabeceraRetiro cabecera = new ECabeceraRetiro();
            cabecera = (ECabeceraRetiro)Session["cabecera"];
            cabecera.IdRetiro = 0;
            cabecera.CedoRuc = TextBoxCedoRuc.Text;
            cabecera.FechaSolicitada = DateTime.Now.ToString();
            cabecera.DiaRetiro = DropDownListDia.SelectedItem.ToString();
            cabecera.HoraRetiro = DropDownListHora.SelectedItem.ToString();
            cabecera.ReferenciaDomicilio = TextBoxReferenciaDomicilio.Text;
            cabecera.Latitud = TextBoxLatitud.Text;
            cabecera.Longitud = TextBoxLongitud.Text;
            cabecera.EstadoRetiro = 1;
            cabecera.EstadoEliminacion = 1;
            Session["mensaje"] = LabelMensaje.Text;
            Session["dt"] = dt;
            Session["cabecera"] = cabecera;
            Response.Redirect("ISolicitarRetiro.aspx", true);
        }
    }
    else

```

**Figura 81. Sección de no ser agregado correctamente**  
**Elaborado por: Valeria Quispe**

```
else
{
    nombre = TextBoxNombreArticulo.Text;
    marca = TextBoxMarcaArticulo.Text;
    modelo = TextBoxModeloArticulo.Text;
    descripcion = TextBoxDescripcionArticulo.Text;
    tipo = DropDownListTipoTrabajo.SelectedItem.ToString();
    dt.Rows.Add(nombre, marca, modelo, descripcion, tipo);
    GridViewDetalle.DataSource = dt;
    GridViewDetalle.DataBind();
    ECabeceraRetiro cabera = new ECabeceraRetiro();
    cabecera = (ECabeceraRetiro)Session["cabecera"];
    cabecera.IdRetiro = 0;
    cabecera.CedoRuc = TextBoxCedoRuc.Text;
    cabecera.FechaSolicitada = DateTime.Now.ToString();
    cabecera.DiaRetiro = DropDownListDia.SelectedItem.ToString();
    cabecera.HoraRetiro = DropDownListHora.SelectedItem.ToString();
    cabecera.ReferenciaDomicilio = TextBoxReferenciaDomicilio.Text;
    cabecera.Latitud = TextBoxLatitud.Text;
    cabecera.Longitud = TextBoxLongitud.Text;
    cabecera.EstadoRetiro = 1;
    cabecera.EstadoEliminacion = 1;
    LabelMensaje.ForeColor = System.Drawing.Color.Green;
    LabelMensaje.Text = "<b>Mensaje</b> : Artículo agregado correctamente";
    limpiarCamposArticulo();

    Session["mensaje"] = LabelMensaje.Text;
    Session["dt"] = dt;
    Session["cabecera"] = cabecera;
    Response.Redirect("ISolicitarRetiro.aspx", true);
}
}
catch (Exception)
{
}
}
```

**Figura 82.** Sección de ser agregado correctamente  
**Elaborado por:** Valeria Quispe

Finalmente se procede a desarrollar el método de validación y registro, para ello el método de ValidarRetiro(), hace una revisión de los datos faltantes y de ser incompleta se presenta al usuario un mensaje “Datos de cabecera faltantes” y en caso de que la información necesaria este completa se procede con el registro de la solicitud.

```

protected void ValidarRetiro()
{
    DataTable dt = new DataTable();
    dt = (DataTable)Session["dt"];
    if (TextBoxCedoRuc.Text == "" || TextBoxReferenciaDomicilio.Text=="")
    {
        LabelMensaje.ForeColor = System.Drawing.Color.DarkBlue;
        LabelMensaje.Text = "<b>Mensaje</b> : Datos de cabecera faltantes";
        ECabeceraRetiro cabera = new ECabeceraRetiro();
        cabecera = (ECabeceraRetiro)Session["cabecera"];
        cabecera.IdRetiro = 0;

        cabecera.CedoRuc = TextBoxCedoRuc.Text;
        cabecera.FechaSolicitada = DateTime.Now.ToString();
        cabecera.DiaRetiro = DropDownListDia.SelectedItem.ToString();
        cabecera.HoraRetiro = DropDownListHora.SelectedItem.ToString();
        cabecera.ReferenciaDomicilio = TextBoxReferenciaDomicilio.Text;
        cabecera.Latitud = TextBoxLatitud.Text;
        cabecera.Longitud = TextBoxLongitud.Text;
        cabecera.EstadoRetiro = 1;
        cabecera.EstadoEliminacion = 1;

        Session["mensaje"] = LabelMensaje.Text;
        Session["dt"] = dt;
        Session["cabecera"] = cabecera;
        Response.Redirect("ISolicitarRetiro.aspx", true);
    } else
    {
        GuardarRetiro();
    }
}

```

**Figura 83.** Método de validación de retiro en la capa de interfaz  
**Elaborado por:** Valeria Quispe

Todo este proceso se lleva a cabo en la capa de interfaz, continuando se desarrolla el método GuardarRetiro(), mismo que recibirá los datos del retiro y este hará el llamado al método de capa de negocio que enviará entidades con la información y este a su vez hará el llamado al método perteneciente a la capa de datos que hará el registro en el servidor de base de datos.

```

private void GuardarRetiro()
{ try{
    ECabeceraRetiro confirm;
    DataTable dt = new DataTable();
    ECabeceraRetiro cabecera = new ECabeceraRetiro();
    String fecha = DateTime.Now.ToShortDateString();
    dt = (DataTable)Session["dt"];
    cabecera = (ECabeceraRetiro)Session["cabecera"];
    string a = TextBoxLatitud.Text;
    confirm = NCabeceraRetiro.GuardarCabeceraRetiro(new ECabeceraRetiro
    {
        IdRetiro = 0,
        IdUsuarioRetiro = Convert.ToInt32(TextBoxUsuario.Text),
        CedoRuc = TextBoxCedoRuc.Text,
        FechaSolicitada = DateTime.Now.ToShortDateString(),
        DiaRetiro = DropDownListDia.SelectedItem.ToString(),
        HoraRetiro = DropDownListHora.SelectedItem.ToString(),
        ReferenciaDomicilio = TextBoxReferenciaDomicilio.Text,
        Latitud = TextBoxLatitud.Text,
        Longitud = TextBoxLongitud.Text,
        EstadoRetiro = 1,
        EstadoEliminacion = 1
    }, dt);
    if (confirm != null) {
        //Guardar Estado Inicializado de Retiro
        EEstadoRetiro estado = new EEstadoRetiro();
        estado.IdEstadoRetiro = 0;
        estado.IdRetiroPertenece = confirm.IdRetiro;
        estado.FechaEstado = confirm.FechaSolicitada;
        estado.EstadoRetiro = "Iniciado";
        estado.IdUsuarioResponsable = confirm.IdUsuarioRetiro;
        estado.NumEstadoRetiro = 1;
        estado = NEstadoRetiro.GuardarEstado(estado);

        Session["mensaje"] = null;
        Session["dt"] = null;
        Session["cabecera"] = null;
        LabelMensaje.Text = "Registro Correcto";
        //Response.Redirect("GestionRecorridos.aspx", true);
    } else{
        LabelMensaje.Text = "Retiro no guardado ";
        Session["mensaje"] = LabelMensaje.Text;
        Session["dt"] = dt;
        Session["cabecera"] = cabecera;
        Response.Redirect("ISolicitarRetiro.aspx", true);
    }
} catch (Exception){
    LabelMensaje.Text = "Registro erróneo, intente más tarde";
}
}

```

**Figura 84.** Método de guardar retiro en la capa de interfaz  
**Elaborado por:** Valeria Quispe

El método de la capa de negocio recibe la información de la capa de interfaz y envía la información a través de entidades a la capa de datos.



```

public class NCabeceraRetiro
{
    public static ECabeceraRetiro GuardarCabeceraRetiro(ECabeceraRetiro cabeceraRetiro, DataTable dt)
    {
        return DCabeceraRetiro.GuardarRetiro(cabeceraRetiro, dt);
    }
}

```

**Figura 85.** Método de guardar retiro de la capa de negocio  
**Elaborado por:** Valeria Quispe

El método guardar retiro de la capa de datos es el responsable de realizar el registro de la información en el servidor de base de datos, este a través de sentencias SQL.

```

public static ECabeceraRetiro GuardarRetiro(ECabeceraRetiro cabeceraRetiro, DataTable dt)
{ try{
    int confirm;
    string query = "INSERT INTO [dbo].[RetiroCabecera]([IdUsuarioRetiro]" +
        ",[CedoRuc],[FechaSolicitada],[DiaRetiro]" +
        ",[HoraRetiro],[ReferenciaDomicilio]" +
        ",[Latitud],[Longitud],[EstadoRetiro],[EstadoEliminacion])VALUES" +
        " (@idusuario," +
        " @ced," +
        " @fecha," +
        " @dia," +
        " @hora," +
        " @domicilio," +
        " @latitud," +
        " @longitud," +
        " @estado," +
        " @eliminacion)" +
        " SELECT SCOPE_IDENTITY()";
    string dtquery = "INSERT INTO [dbo].[RetiroDetalle]([NombreArticulo]" +
        ",[MarcaArticulo],[ModeloArticulo],[DescripcionArticulo]" +
        ",[TipoServicio],[IdRetiroPertenece])VALUES" +
        " (@articulo," +
        " @marca," +
        " @modelo," +
        " @descripcion," +
        " @tipo," +
        " @retirooper)";
    using (SqlConnection connection = new SqlConnection((Settings1.Default.Conexion)))

```

```

{
    connection.Open();
    var transaccion = connection.BeginTransaction();
    using (SqlCommand cmd = new SqlCommand(query, connection, transaccion))
    {
        cmd.Parameters.AddWithValue("@idusuario", cabeceraRetiro.IdUsuarioRetiro);
        cmd.Parameters.AddWithValue("@ced", cabeceraRetiro.CedoRuc);
        cmd.Parameters.AddWithValue("@fecha", cabeceraRetiro.FechaSolicitada);
        cmd.Parameters.AddWithValue("@dia", cabeceraRetiro.DiaRetiro);
        cmd.Parameters.AddWithValue("@hora", cabeceraRetiro.HoraRetiro);
        cmd.Parameters.AddWithValue("@domicilio", cabeceraRetiro.ReferenciaDomicilio);
        cmd.Parameters.AddWithValue("@latitud", cabeceraRetiro.Latitud);
        cmd.Parameters.AddWithValue("@longitud", cabeceraRetiro.Longitud);
        cmd.Parameters.AddWithValue("@estado", cabeceraRetiro.EstadoRetiro);
        cmd.Parameters.AddWithValue("@eliminacion", cabeceraRetiro.EstadoEliminacion);
        cabeceraRetiro.IdRetiro = Convert.ToInt32(cmd.ExecuteScalar());
        if (cabeceraRetiro.IdRetiro > 0)
            confirm = 1;
        else
            confirm = -1;
    }
    using (SqlCommand cmd = new SqlCommand(dtquery, connection, transaccion))
    {
        foreach (DataRow item in dt.Rows) {
            cmd.Parameters.Clear();
            cmd.Parameters.AddWithValue("@articulo", item["Nombre"]);
            cmd.Parameters.AddWithValue("@marca", item["Marca"]);
            cmd.Parameters.AddWithValue("@modelo", item["Modelo"]);
            cmd.Parameters.AddWithValue("@descripcion", item["Descripcion"]);
            cmd.Parameters.AddWithValue("@tipo", item["Tipo"]);
            cmd.Parameters.AddWithValue("@retirooper", cabeceraRetiro.IdRetiro);
            cmd.ExecuteNonQuery();
        }
        transaccion.Commit();
        connection.Close();
    }
    return cabeceraRetiro;
}
catch (Exception){return null;
}
}

```

**Figura 86.** Método de guardar retiro de la capa de datos  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Estado de retiros


El formulario de estado de retiros presenta al cliente un listado de los retiros realizados y que no se encuentran finalizados en su totalidad, con esto podrá acceder al detalle del retiro.




Id Retiro	Fecha Solicitada	Día Estimado	Hora Estimada	Referencia del Domicilio	Ver Estado
1	29/03/2022	Jueves	15:00	Frente de upc	
2	29/03/2022	Miercoles	16:00	Frente de upc	
3	29/03/2022	Jueves	17:00	Frente de upc	
4	30/03/2022	Miercoles	16:00	Frente de upc	

**Figura 87.** Formulario de visualizar estado de retiro  
**Elaborado por:** Valeria Quispe

Al visualizar el detalle del retiro podrá verificar la información de los dispositivos y los estados por el cual ha sido manipulado, con ello podrá ver y constatar en qué estado se encuentra su operación.



**EPC-COMPU**  
 RICARDO PÉREZ GERENTE  
 Dirección: La Quigo-Atocha  
 Mantenimiento y Reparación de Equipos Electrónicos



---

**ESTADOS DEL RETIRO N°3**

**INFORMACIÓN DEL CLIENTE**

Nombre:	Valeria Quispe	RUC / CI:	1805153580
Dirección:	Ciudadela curamunda Alrededor	Fecha Solicitada:	29/03/2022
Teléfono:	0979398197	Día y Hora estimada:	Jueves-17:00
Referencia:	Frete de upc	Retiro:	3

**ARTÍCULOS DEL RETIRO**

Nombre	Marca	Modelo	Descripción	Servicio
CPU	DELL	C.00	Manuales	Mantenimiento

**ESTADOS DEL RETIRO**

Id Estado	Fecha	Estado
3	07/04/2022	Proceso
1002	13/04/2022	Ingresado
1003	18/04/2022	Revisión
1004	25/04/2022	Diagnosticado

**DESCRIPCIÓN DE ESTADOS**

Estado	Descripción
Iniciado	Cliente solicito el retiro a domicilio de los dispositivos
Proceso	Personal de la empresa se comunicará con el cliente para realizar el retiro en el domicilio
Ingresado	Personal de la empresa aceptará los dispositivos recogidos del domicilio
Revisión	Personal técnico se encuentra en revisión de los equipos tecnológicos
Diagnosticado	Personal de la empresa emite la orden de Reparación o Mantenimiento al cliente
Completado	El cliente aceptará o rechazará realizar el trabajo, empezando el proceso de Reparación

22/06/2022
Informe emitido por Sistema Web EPC-Compu

**Figura 88.** Informe de historial de estado de un retiro  
**Elaborado por:** Valeria Quispe

### Codificación

En la presentación de los retiros realizados cuyos estados se encuentran pendientes se desarrolla el método RecuperarRetiros(), este a su vez hará el llamado al método de la capa de negocio y así hasta llegar a la capa de datos, todo a favor de recuperar los datos de gestor de base de datos.

```

private void RecuperarRetiros()
{
    List<ECabeceraRetiro> listaRetiros = new List<ECabeceraRetiro>();
    listaRetiros = NCabeceraRetiro.ObtenerRetirosUsuario(usuario.IdUsuario);
    List<ECabeceraRetiro> listaRetirosFecha = new List<ECabeceraRetiro>();
    if (listaRetiros == null)
    {
    }
    else
    {
        foreach (var item in listaRetiros)
        {
            ECabeceraRetiro retiro = new ECabeceraRetiro();
            retiro.IdRetiro = item.IdRetiro;
            retiro.DiaRetiro = item.DiaRetiro;
            retiro.HoraRetiro = item.HoraRetiro;
            retiro.ReferenciaDomicilio = item.ReferenciaDomicilio;
            DateTime fecha = Convert.ToDateTime(item.FechaSolicitada);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            retiro.FechaSolicitada = fecha1;
            listaRetirosFecha.Add(retiro);
        }

        GridViewRetiros.DataSource = listaRetirosFecha;
        GridViewRetiros.DataBind();
    }
}

```

**Figura 89.** Método de recuperar retiros en la capa de interfaz  
**Elaborado por:** Valeria Quispe

El método `ObtenerRetirosUsuarios()`, se encargará de recuperar los retiros pendientes por usuario esto a través de sentencias SQL.

```

public static List<ECabeceraRetiro> ObtenerRetirosUsuario(int id)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdRetiro]
, [IdUsuarioRetiro]
, [CedoRuc]
, [FechaSolicitada]
, [DiaRetiro]
, [HoraRetiro]
, [ReferenciaDomicilio]
, [Latitud]
, [Longitud]
, [EstadoRetiro]
, [EstadoEliminacion]
FROM [dbo].[RetiroCabecera]
        where IdUsuarioRetiro=" + id + """;
        cmd.CommandType = System.Data.CommandType.Text;
        List<ECabeceraRetiro> listaDetalle = new List<ECabeceraRetiro>();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                ECabeceraRetiro detalle = new ECabeceraRetiro();
                detalle.IdRetiro= Convert.ToInt32(dr["IdRetiro"].ToString());
                detalle.IdUsuarioRetiro = Convert.ToInt32(dr["IdUsuarioRetiro"].ToString());
                detalle.CedoRuc = dr["CedoRuc"].ToString();
                detalle.FechaSolicitada = (dr["FechaSolicitada"].ToString());
                detalle.DiaRetiro = (dr["DiaRetiro"].ToString());
                detalle.HoraRetiro = (dr["HoraRetiro"].ToString());
                detalle.ReferenciaDomicilio = (dr["ReferenciaDomicilio"].ToString());
                detalle.Latitud = (dr["Latitud"].ToString());
                detalle.Longitud = (dr["Longitud"].ToString());
                detalle.EstadoRetiro = Convert.ToInt32(dr["EstadoRetiro"].ToString());
                detalle.EstadoEliminacion = Convert.ToInt32(dr["EstadoEliminacion"].ToString());
                listaDetalle.Add(detalle);
            }
        }
        conexion.Close();
        return listaDetalle;
    }
    catch (Exception)
    {
        return null;
    }
}

```

**Figura 90.** Método de recuperar retiros por usuario en la capa de datos

**Elaborado por:** Valeria Quispe

Para completar con la visualización del informe se realiza el método para la recolección de los datos y a su vez se realiza el diseño del informe con la combinación de html.

```

protected void generarPDFEstado(int idRetiro)
{
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(idRetiro);
    DateTime fechaSolicitada = Convert.ToDateTime(retiro.FechaSolicitada);
    string fechaSol = fechaSolicitada.ToString("dd/MM/yyyy");
    EUsuario usuario = NUsuario.DevolverUsuarioxID(retiro.IdUsuarioRetiro);
    EPersona persona = NPersona.DevolverPersonaxID(usuario.IdPersonaPertenece);
    List<EDetalleRetiro> listaDetalleRetiro = NCabeceraRetiro.ObtenerDetalleRetiroxID(retiro.IdRetiro);
    string detalleRetiro = "";
    foreach (var item in listaDetalleRetiro)
    {
        detalleRetiro += "<tr><td style='width:18%;>" + item.NombreArticulo + "</td><td style='width:18%;>" + item.MarcaArticulo + "</td>" +
            "<td style='width:18%;>" + item.ModeloArticulo + "</td><td style='width:24%;>" + item.DescripcionArticulo + "</td>" +
            "<td style='width:18%;>" + item.TipoServicio + "</td></tr>";
    }

    List<EEstadoRetiro> listaDetalleEstados = NEstadoRetiro.DevolverEstadosxIDRetiro(retiro.IdRetiro);
    string detalleEstado = "";

    foreach (var item in listaDetalleEstados)
    {
        DateTime fechaM = Convert.ToDateTime(item.FechaEstado);
        string fecha1 = fechaM.ToString("dd/MM/yyyy");
        detalleEstado += "<tr><td style='width:33%;>" + item.IdEstadoRetiro + "</td>" +
            "<td style='width:33%;>" + fecha1 + "</td><td style='width:33%;>" + item.EstadoRetiro + "</td></tr>";
    }
    string fecha = DateTime.Now.ToShortDateString();
    string salto = DevolverSalto(listaDetalleRetiro, listaDetalleEstados);
}

```

```

string html = "<!DOCTYPE html>" +
    "<html><head><meta charset='UTF-8'><title> Diagnóstico del Retiro N°" + retiro.IdRetiro + " </title>" +
    "</head><style>.class_table {border-collapse: separate;border-spacing: 10px;border: 1px solid black;border-radius: 15px;-moz-border-radius: 20px;padding: 2px}</style><body (height: 500px;background-image:url('');background-repeat:no-repeat;background-position-x:center;background-position-y:250px;background-size: 30rem)</body><table border='1'><tr><td colspan='3'><img alt='Logo de la empresa' src='https://i.ibb.co/mq01rtd/Logo-Ambato.jpg' width='100px' height='100px'></td></tr></table></body></html>";
}

var pdf = Pdf
    .From(html)
    .WithGlobalSetting("orientation", "portrait")
    .WithObjectSetting("web.defaultEncoding", "utf-8")
    .Content();

Session["download"] = pdf;
Response.Redirect("IDescargarEstado.aspx", true);
}

```

**Figura 91.** Método de generar informe en la capa de interfaz  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Asignación de retiros

La asignación de retiros se enfoca en que el personal pertinente pueda visualizar las asignaciones de retiros pendientes, con ello facilitará la organización y la gestión del servicio ofrecido, en el mismo se presenta el nombre del cliente, dirección, fecha de asignación realizada y el personal que fue encargado para realizar la recolección de los dispositivos tecnológicos.

### Asignación de Retiros

LISTADO DE ASIGNACIONES DE RETIROS 👤

Id Asignación	Cliente	DireccionCliente	Fecha de Asignación	Responsable de Retiro
1	Valeria Quispe	Ciudadela cumanda Alrededor	06/04/2022	Fernando Naula

Agregar Asignación

**Figura 92.** Vista de asignaciones de retiros pendientes  
**Elaborado por:** Valeria Quispe

Si el usuario deseará ingresar una nueva asignación de retiro, se le presentará el formulario de la figura, donde tendrá que seleccionar el retiro de interés y el personal encargado de la recolección. En caso de que el registro sea satisfactorio se le direccionará a la vista de asignaciones pendientes y si no se presentará un mensaje al usuario.

### Asignación de Retiros

#### Registrar Asignación

DATOS DEL RETIRO 🕒

Id Retiro:	<input type="text"/>	<span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 4px;">Buscar</span>	Nombre del Cliente:	<input type="text"/>
Dirección:	<input type="text"/>		Teléfono:	<input type="text"/>

DATOS DEL PERSONAL ENCARGADO 👤

Id Usuario:	<input type="text"/>	<span style="background-color: #007bff; color: white; padding: 2px 5px; border-radius: 4px;">Buscar</span>	Nombre de Personal:	<input type="text"/>
Dirección:	<input type="text"/>		Teléfono:	<input type="text"/>

Guardar
Cancelar

**Figura 93.** Formulario de registrar asignación de retiros  
**Elaborado por:** Valeria Quispe

La ventana emergente de retiros solicitados con el estado pendientes cumple con la finalidad de que el usuario pueda seleccionar el retiro de interés a asignar para su recolección, en ello podrá visualizar el nombre del cliente, fecha solicitada, día y hora estimada y dirección del domicilio.



**Figura 94.** Vista de retiros solicitados pendientes  
**Elaborado por:** Valeria Quispe

La ventana emergente de personal de entrega y recepción cumple con la finalidad de que el usuario pueda seleccionar el usuario que se encargará de realizar la recolección de los dispositivos tecnológicos en el domicilio del cliente.



**Figura 95.** Vista de personal de entrega y recepción  
**Elaborado por:** Valeria Quispe

Codificación



Primero se realiza el método `CargarListadoAsignaciones()`, encargado de presentar el listado de asignaciones de estado pendientes, dicho método solicita el llamado a otro método de la capa de negocio y este a otro de la capa de datos para obtener los datos del servidor de base de datos. Las operaciones de solicitud se lo hacen a través del envío y recepción de información este en forma de entidades

```
private void CargarListadoAsignaciones()
{
    int estado = 1;
    List<EAsignarRetiro> listaAsignaciones = new List<EAsignarRetiro>();
    listaAsignaciones = NAsignarRetiro.ObtenerAsignacionesxEstado(estado);
    List<EVistaAsignarRetiro> listaVista = new List<EVistaAsignarRetiro>();
    if (listaAsignaciones == null)
    {
    }
    else
    {
        foreach (var item in listaAsignaciones)
        {
            ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(item.IdRetiroAsignacion);
            EUsuario cliente = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
            EPersona perCliente = NPersona.DevolverPersonaxId(cliente.IdPersonaPertenece);

            EUsuario responsable = NUsuario.DevolverUsuarioxId(item.IdUsuarioAsignacion);
            EPersona perResponsable = NPersona.DevolverPersonaxId(responsable.IdPersonaPertenece);
            EVistaAsignarRetiro vista = new EVistaAsignarRetiro();

            vista.IdAsignacionRetiro = item.IdAsignacionRetiro;
            vista.Cliente = perCliente.NombrePersona + " " + perCliente.ApellidoPersona;
            vista.DireccionCliente = perCliente.DireccionPersona;
            DateTime fecha = Convert.ToDateTime(item.FechaAsignacion);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaAsignacion = fecha1;
            vista.ResponsableAsignacion = perResponsable.NombrePersona + " " + perResponsable.ApellidoPersona;
            listaVista.Add(vista);
        }
    }
    GridViewAsignaciones.DataSource = listaVista;
    GridViewAsignaciones.DataBind();
}
```

**Figura 96.** Método de visualizar asignaciones de retiros en la capa de interfaz  
**Elaborado por:** Valeria Quispe

El llamado a los métodos se lo hace de la siguiente forma con el fin de poder solicitar la información requerida, mismo en forma de entidades.

```
ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(item.IdRetiroAsignacion);
EUsuario cliente = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
EPersona perCliente = NPersona.DevolverPersonaxId(cliente.IdPersonaPertenece);

EUsuario responsable = NUsuario.DevolverUsuarioxId(item.IdUsuarioAsignacion);
EPersona perResponsable = NPersona.DevolverPersonaxId(responsable.IdPersonaPertenece);
```

**Figura 97.** Llamado a métodos de la capa de negocio  
**Elaborado por:** Valeria Quispe

Los métodos de la capa de negocio cumplen su intermedio entre la capa de datos e interfaz y dichos métodos son llamados como se presenta en la figura.

```

public static List<EAsignarRetiro> ObtenerAsignacionesxEstado(int estado)
{
    return DAsignarRetiro.ObtenerAsignacionesxEstado(estado);
}

```

**Figura 98.** Método de obtener asignaciones de retiros pendientes en la capa de negocio  
**Elaborado por:** Valeria Quispe

El método que se encuentra en la capa de datos cumple con la finalidad de recuperar la información del servidor de datos, mismos que son tratados y transformados en entidades para su previa utilización en las diferentes capas, la recuperación se lo realiza a través de las sentencias SQL que se comunican por la conexión realizada con el servidor de base de datos.

```

public static List<EAsignarRetiro> ObtenerAsignacionesxEstado(int estado)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdAsignacionRetiro]
        ,[IdRetiroAsignacion]
        ,[IdUsuarioAsignacion]
        ,[FechaAsignacion]
        ,[EstadoAsignacion]
        FROM [dbo].[AsignacionRetiro] where EstadoAsignacion=" + estado + """;
        cmd.CommandType = System.Data.CommandType.Text;
        List<EAsignarRetiro> listaDetalle = new List<EAsignarRetiro>();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                EAsignarRetiro detalle = new EAsignarRetiro();
                detalle.IdAsignacionRetiro = Convert.ToInt32(dr["IdAsignacionRetiro"].ToString());
                detalle.IdRetiroAsignacion = Convert.ToInt32(dr["IdRetiroAsignacion"].ToString());
                detalle.IdUsuarioAsignacion = Convert.ToInt32(dr["IdUsuarioAsignacion"].ToString());
                detalle.FechaAsignacion = (dr["FechaAsignacion"].ToString());
                detalle.EstadoAsignacion = Convert.ToInt32(dr["EstadoAsignacion"].ToString());
                listaDetalle.Add(detalle);
            }
        }
        conexion.Close();
        return listaDetalle;
    } catch (Exception)
    {
        return null;
    }
}

```

**Figura 99.** Método de obtener asignaciones de retiros pendientes en la capa de datos  
**Elaborado por:** Valeria Quispe

Conforme a cumplir con toda la funcionalidad se lleva a cabo el desarrollo del método GuardarAsignacion(), para realizar el registro de la asignación. Este a su vez

hará el llamado a diferentes métodos para cumplir de forma correcta la operación y de ser errónea se presentará un mensaje al usuario.

```
private void GuardarAsignacion()
{
    EAsignarRetiro asignacion = new EAsignarRetiro();
    asignacion.IdAsignacionRetiro = 0;
    asignacion.IdRetiroAsignacion = Convert.ToInt32(LabelIdRetiro.Text);
    asignacion.IdUsuarioAsignacion = Convert.ToInt32(LabelIdPersonal.Text);
    asignacion.EstadoAsignacion = 1;
    asignacion.FechaAsignacion = DateTime.Now.ToShortDateString();
    asignacion = NAsignarRetiro.GuardarAsignacionRetiro(asignacion);

    if (asignacion == null)
    {
        LabelMensaje.Text = "Registro no registrado correctamente";
    }
    else
    {
        GenerarEstadoProceso(asignacion);
        ActualizarEstadoRetiro(asignacion);
        LabelMensajeGrid.Text = "Registrado correctamente";
        CargarListadoPersonalDisponible();
        CargarListadoRetirosIniciados();
        limpiarCampos();
        CargarListadoAsignaciones();
        MultiView1.ActiveViewIndex = 1;
    }
}
```

**Figura 100.** Método de guardar asignación de retiro en la capa de interfaz

**Elaborado por:** Valeria Quispe

El siguiente método recepta la entidad asignación que a su vez la información en envía al servidor de base de datos mediante la sentencia SQL, INSERT.

```

public static EAsignarRetiro GuardarAsignacionRetiro(EAsignarRetiro asignacion)
{
    try
    {
        SqlConnection connection = new SqlConnection(Settings1.Default.Conexion);
        connection.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = connection;
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = @"INSERT INTO [dbo].[AsignacionRetiro]
([IdRetiroAsignacion]
,[IdUsuarioAsignacion]
,[FechaAsignacion]
,[EstadoAsignacion])
VALUES
(@retiro,@usuario,@fecha,@estado);SELECT SCOPE_IDENTITY()";
        cmd.Parameters.AddWithValue("@retiro", asignacion.IdRetiroAsignacion);
        cmd.Parameters.AddWithValue("@usuario", asignacion.IdUsuarioAsignacion);
        cmd.Parameters.AddWithValue("@fecha", asignacion.FechaAsignacion);
        cmd.Parameters.AddWithValue("@estado", asignacion.EstadoAsignacion);
        var idAsignacion = Convert.ToInt32(cmd.ExecuteScalar());
        asignacion.IdAsignacionRetiro = idAsignacion;
        connection.Close();
        return asignacion;
    } catch (Exception)
    {
        return null;
    }
}

```

**Figura 101.** Método de guardar asignación de retiro en la capa de datos  
**Elaborado por:** Valeria Quispe

El desarrollo de la vista de retiros solicitados con el estado pendiente, es llevado a cabo mediante la realización de CargarListadoRetirosIniciados(), que cumple con la recuperación de la información en capa de interfaz.

```

private void CargarListadoRetirosIniciados()
{
    int estado = 1;
    List<ECabeceraRetiro> listaRetirosIniciados = NCabeceraRetiro.ObtenerRetirosxEstado(estado);
    List<EVistaRetirosIniciado> listaVistaRetiros = new List<EVistaRetirosIniciado>();
    if (listaRetirosIniciados == null)
    {
    }
    else
    {
        foreach (var item in listaRetirosIniciados)
        {
            EVistaRetirosIniciado vista = new EVistaRetirosIniciado();
            EUsuario usuario = NUsuario.DevolverUsuarioId(item.IdUsuarioRetiro);
            EPersona persona = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);
            vista.IdRetiro = item.IdRetiro;
            vista.NombreCliente = persona.NombrePersona + " " + persona.ApellidoPersona;
            DateTime fecha = Convert.ToDateTime(item.FechaSolicitada);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaSolicitada = fecha1;
            vista.DiaHoraRetiro = item.DiaRetiro + " " + item.HoraRetiro;
            vista.Domicilio = persona.DireccionPersona;
            vista.Ubicacion = "Coord:(" + item.Latitud + "," + item.Longitud + ")";
            listaVistaRetiros.Add(vista);
        }
        GridViewRetiros.DataSource = listaVistaRetiros;
        GridViewRetiros.DataBind();
    }
}

```

**Figura 102.** Método de retiros pendientes en la capa de interfaz

**Elaborado por:** Valeria Quispe

El método de obtención de retiros pendientes en la capa de datos es el encargado de hacer la gestión de recuperación de información de los retiros, mismos que son transformados en entidades para su previa utilización en las demás capas.

```
public static List<ECabeceraRetiro> ObtenerRetirosxEstado(int estado)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdRetiro]
        ,[IdUsuarioRetiro],[CedoRuc]
        ,[FechaSolicitada],[DiaRetiro]
        ,[HoraRetiro],[ReferenciaDomicilio]
        ,[Latitud],[Longitud]
        ,[EstadoRetiro],[EstadoEliminacion]
        FROM [dbo].[RetiroCabecera] where EstadoRetiro=" + estado + """;
        cmd.CommandType = System.Data.CommandType.Text;
        List<ECabeceraRetiro> listaDetalle = new List<ECabeceraRetiro>();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                ECabeceraRetiro detalle = new ECabeceraRetiro();
                detalle.IdRetiro = Convert.ToInt32(dr["IdRetiro"].ToString());
                detalle.IdUsuarioRetiro = Convert.ToInt32(dr["IdUsuarioRetiro"].ToString());
                detalle.CedoRuc = dr["CedoRuc"].ToString();
                detalle.FechaSolicitada = (dr["FechaSolicitada"].ToString());
                detalle.DiaRetiro = (dr["DiaRetiro"].ToString());
                detalle.HoraRetiro = (dr["HoraRetiro"].ToString());
                detalle.ReferenciaDomicilio = (dr["ReferenciaDomicilio"].ToString());
                detalle.Latitud = (dr["Latitud"].ToString());
                detalle.Longitud = (dr["Longitud"].ToString());
                detalle.EstadoRetiro = Convert.ToInt32(dr["EstadoRetiro"].ToString());
                detalle.EstadoEliminacion = Convert.ToInt32(dr["EstadoEliminacion"].ToString());
                listaDetalle.Add(detalle);
            }
        }
        conexion.Close();
        return listaDetalle;
    }
    catch (Exception)
    {
        return null;
    }
}
```

**Figura 103.** Método de recuperar retiros pendientes en la capa de datos

**Elaborado por:** Valeria Quispe

El desarrollo de la vista de personal de entrega y recepción, es llevado a cabo mediante la realización de CargarListadoPersonalDisponibel(), que cumple con la recuperación de la información en capa de interfaz.

```

private void CargarListadoPersonalDisponible()
{
    int perfil = 5;
    List<EUsuario> listaUsuariosRE = NUsuario.DevolverUsuariosxPerfil(perfil);
    List<EPersona> listaPersonaRE = new List<EPersona>();
    if (listaUsuariosRE == null)
    {
    }
    else
    {
        foreach (var item in listaUsuariosRE)
        {
            EPersona persona = NPersona.DevolverPersonaxId(item.IdPersonaPertenece);
            listaPersonaRE.Add(persona);
        }
        GridViewPersonal.DataSource = listaPersonaRE;
        GridViewPersonal.DataBind();
    }
}
}

```

**Figura 104.** Método de recuperar de personal de entrega y recepción en la capa de interfaz

**Elaborado por:** Valeria Quispe

El método de obtención de personal de entrega y recepción en la capa de datos es el encargado de hacer la gestión de recuperación de información de los usuarios con dicho perfil, mismos son transformados en entidades para su previa utilización en las demás capas.

```

public static List<EUsuario> DevolverUsuariosxPerfil(int v)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdUsuario]
        ,[ContraseniaUsuario]
        ,[IdPerfilPertenece]
        ,[IdPersonaPertenece]
        FROM [dbo].[Usuario]
        where IdPerfilPertenece=@id";
        cmd.Parameters.AddWithValue("@id", v);
        cmd.CommandType = System.Data.CommandType.Text;
        List<EUsuario> listaUsuarios = new List<EUsuario>();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                EUsuario usuario = new EUsuario();
                usuario.IdUsuario = Convert.ToInt32(dr["IdUsuario"].ToString());
                usuario.ContraseniaUsuario = dr["ContraseniaUsuario"].ToString();
                usuario.IdPerfilPertenece = Convert.ToInt32(dr["IdPerfilPertenece"].ToString());
                usuario.IdPersonaPertenece = Convert.ToInt32(dr["IdPersonaPertenece"].ToString());
                listaUsuarios.Add(usuario);
            }
        }
        conexion.Close();
        return listaUsuarios;
    }
    catch (Exception)
    {
        return null;
    }
}

```

**Figura 105.** Método de recuperar personal de entrega y recepción en la capa de datos

**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Ver Asignación

La visualización de asignación de retiros presenta un listado de retiros asignados a un usuario con el perfil personal de entrega y recepción y se encuentran en un estado pendiente, esto con el fin de ayudar en su organización para realizar la recepción de los dispositivos durante los diferentes días presentando información como, nombre del cliente, dirección y fecha de la asignación. A su vez presenta una opción de poder visualizar el detalle de la asignación.

## Visualización de Asignación de Retiros

LISTADO DE RETIROS ASIGNADOS				
Id Asignación	Cliente	Dirección del Cliente	Fecha de Asignación	Ver Asignación
1	Valeria Quispe	Ciudadela cumanda Alrededor	06/04/2022	

**Figura 106.** Vista de retiros asignados al personal de entrega y recepción

**Elaborado por:** Valeria Quispe

La visualización de asignación presenta un detallado de la información del cliente, resaltando la ubicación del domicilio y dispositivos a retirar. Es importante destacar que se presenta un mapa mediante la utilización del api del Google Maps en ello se visualiza la ruta que puede tomar el personal para transportarse del punto de origen al punto del cliente.

## Visualización de Asignación

INFORMACIÓN DEL RETIRO			
Id Retiro:	1	Nombre del Cliente:	Valeria Quispe
Dirección:	Ciudadela cumanda Alrededor	Teléfono:	0979398197
Día - Hora Solicitado:	Jueves-15:00	Referencia Domicilio:	Frente de upc

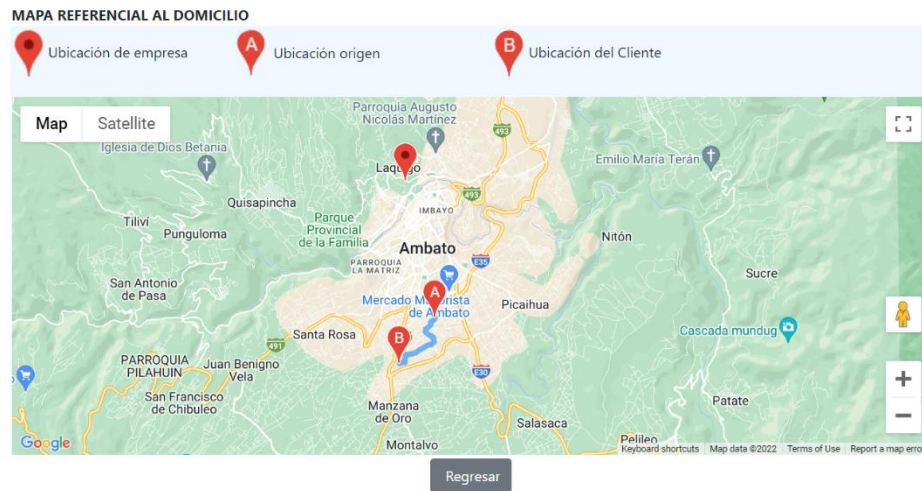
  

DETALLE DE DISPOSITIVOS TECNOLÓGICOS				
Id Detalle	Nombre	Marca	Modelo	Descripción
1	CPU	DELL	del-09	Ninguna
2	Impresora	EPSON	C.00	Manuales

**Figura 107.** Visualización de Asignación

**Elaborado por:** Valeria Quispe





**Figura 108.** Visualización de detalle de asignación de retiro  
**Elaborado por:** Valeria Quispe

### Codificación

Conforme a presentar el listado de asignaciones a cargo del personal de entrega y recepción se lleva a realizar el método de CargarListadoAsignaciones(id usuario), con el fin de recuperar una lista que es emitida por el llamado al método de la capa de negocio y el mismo que hará un llamado al método de la capa de datos. Este encargado de interactuar con el servidor de base de datos.

```
private void CargarListadoAsignaciones(int idUsuario)
{
    int estado = 1;
    List<EAsignarRetiro> listaAsignaciones = new List<EAsignarRetiro>();
    listaAsignaciones = NAsignarRetiro.ObtenerAsignacionesxEstadoIdUsuario(estado, idUsuario);
    List<EVistaAsignarRetiro> listaVista = new List<EVistaAsignarRetiro>();
    if (listaAsignaciones == null)
    {
    }
    else
    {
        foreach (var item in listaAsignaciones)
        {
            ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(item.IdRetiroAsignacion);
            EUsuario cliente = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
            EPersona perCliente = NPersona.DevolverPersonaxId(cliente.IdPersonaPertenece);
            EUsuario responsable = NUsuario.DevolverUsuarioxId(item.IdUsuarioAsignacion);
            EPersona perResponsable = NPersona.DevolverPersonaxId(responsable.IdPersonaPertenece);
            EVistaAsignarRetiro vista = new EVistaAsignarRetiro();
            vista.IdAsignacionRetiro = item.IdAsignacionRetiro;
            vista.Cliente = perCliente.NombrePersona + " " + perCliente.ApellidoPersona;
            vista.DireccionCliente = perCliente.DireccionPersona;
            DateTime fecha = Convert.ToDateTime(item.FechaAsignacion);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaAsignacion = fecha1;
            vista.ResponsableAsignacion = perResponsable.NombrePersona + " " + perResponsable.ApellidoPersona;
            listaVista.Add(vista);
        }
        GridViewAsignaciones.DataSource = listaVista;
        GridViewAsignaciones.DataBind();
    }
}
```

**Figura 109.** Método de obtener asignaciones de retiros por usuario  
pendientes en la capa de interfaz  
**Elaborado por:** Valeria Quispe

El método perteneciente a la capa de datos trata de recuperar el listado de asignaciones pendientes con un parámetro que es el id del usuario, la información recuperada es transformada a las entidades pertinentes para su uso previo uso en las diferentes capas.

```

public static List<EAsignarRetiro> ObtenerAsignacionesxEstadoIdUsuario(int estado, int idUsuario)
{
    try
    {
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdAsignacionRetiro]
        ,[IdRetiroAsignacion]
        ,[IdUsuarioAsignacion]
        ,[FechaAsignacion]
        ,[EstadoAsignacion]
        FROM [dbo].[AsignacionRetiro]
        where EstadoAsignacion=" + estado + "and IdUsuarioAsignacion="+idUsuario+"";

        cmd.CommandType = System.Data.CommandType.Text;
        List<EAsignarRetiro> listaDetalle = new List<EAsignarRetiro>();
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                EAsignarRetiro detalle = new EAsignarRetiro();
                detalle.IdAsignacionRetiro = Convert.ToInt32(dr["IdAsignacionRetiro"].ToString());
                detalle.IdRetiroAsignacion = Convert.ToInt32(dr["IdRetiroAsignacion"].ToString());
                detalle.IdUsuarioAsignacion = Convert.ToInt32(dr["IdUsuarioAsignacion"].ToString());
                detalle.FechaAsignacion = (dr["FechaAsignacion"].ToString());
                detalle.EstadoAsignacion = Convert.ToInt32(dr["EstadoAsignacion"].ToString());
                listaDetalle.Add(detalle);
            }
        }
        conexion.Close();
        return listaDetalle;
    } catch (Exception)
    {
        return null;
    }
}

```

**Figura 110.** Método de obtener asignaciones de retiros por usuario pendientes en la capa de datos  
**Elaborado por:** Valeria Quispe

Para llevar a cabo la visualización de los detalles del retiro se genera el método EnviarDatosRetiro(), que cumple con el envío del id de retiro de interés a la página IRutaDetalleRetiro.aspx

```

private void EnviarDatosRetiro(GridViewCommandEventArgs e)
{
    int index = Convert.ToInt32(e.CommandArgument);
    GridViewRow row = GridViewAsignaciones.Rows[index];
    int IdAsignacionRetiro = Convert.ToInt32(row.Cells[0].Text);
    Session["AsignacionRetiro"] = IdAsignacionRetiro;
    Response.Redirect("IRutaDetalleRetiro.aspx", true);
}

```

**Figura 111.** Método para enviar datos de retiro  
**Elaborado por:** Valeria Quispe

En la página se presenta la información del retiro de interés, para ello se lleva el cabo el desarrollo de métodos como CargarRetiro().

```

private void cargarRetiro(int id)
{
    EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxId(id);
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);
    Session["idRetiro"] = retiro.IdRetiro;
    List<EDetalleRetiro> listadoArticulos = NCabeceraRetiro.ObtenerDetalleRetiroxID(retiro.IdRetiro);
    EUsuario usuario = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
    EPersona cliente = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);

    TextBoxIdRetiro.Text = retiro.IdRetiro.ToString();
    TextBoxDireccion.Text = cliente.DireccionPersona;
    TextBoxDiaHora.Text = retiro.DiaRetiro + "-" + retiro.HoraRetiro;
    TextBoxNombreCliente.Text = cliente.NombrePersona + " " + cliente.ApellidoPersona;
    TextBoxTelefono.Text = cliente.TelefonoPersona;
    TextBoxReferenciaDomi.Text = retiro.ReferenciaDomicilio;

    GridViewDispositivos.DataSource=listadoArticulos;
    GridViewDispositivos.DataBind();
}

```

**Figura 112.** Método de recuperar retiro por id en la capa de interfaz  
**Elaborado por:** Valeria Quispe

A su vez se desarrolla el método para poder recuperar las coordenadas geográficas del cliente, este para ser de uso en la presentación del mapa de ruta entre el punto de origen del personal de entrega y recepción y el cliente, previo a la recolección de los dispositivos tecnológicos para su diagnóstico.

```

public string RecuperarValoresPrimerPosicion()
{
    DataTable dt = new DataTable();

    int id =Convert.ToInt32(Session["idRetiro"]);
    string datos;
    datos = "";
    string sql = (string)Session["Conexion"];
    using (SqlConnection con = new SqlConnection(sql))
    {
        using (SqlCommand cmd = new SqlCommand("SELECT title=retiro.IdRetiro,lat=retiro.Latitud,lng=retiro.Longitud " +
            "FROM RetiroCabecera as retiro WHERE retiro.IdRetiro ='+id+';", con))
        {
            con.Open();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            da.Fill(dt);
            foreach (DataRow col in dt.Rows)
            {
                // datos = datos + "{";
                datos = datos + col[1] + "," + col[2];
                //datos = datos + "},";
            }
            con.Close();
        }
    }
    return datos;
}

```

**Figura 113.** Método de recuperar coordenada geográfica en la capa de datos  
**Elaborado por:** Valeria Quispe

Consigno se realiza el método utilizado para el tratamiento y manipulación de las coordenadas geográficas recuperadas, esto en el trazado de la ruta origen – destino. El mismo es presentado en un mapa que el usuario pueda interpretar la información expuesta y que ayude a organizar la recolección de los dispositivos tecnológicos.

```

<script type="text/javascript">
var mapa = document.getElementById('modal');
navigator.geolocation.getCurrentPosition(fn_ok, fn_mal);
function fn_mal() {
}
function fn_ok(rta) {
var lat = rta.coords.latitude;
var lon = rta.coords.longitude;
var glatlondestino = new google.maps.LatLng(«%=RecuperarValoresPrimerPosicion()%»);
var glatlonEmpresa = new google.maps.LatLng(-1.228611,-78.631347);
var glatlon = new google.maps.LatLng(lat, lon);
var objectCon = {
zoom: 1,
center: glatlon
}
var gMapa = new google.maps.Map(mapa, objectCon);
var marker = {
position: glatlonEmpresa,
map: gMapa,
title: "Local EPC-COMPU"
}
var gmarker = new google.maps.Marker(marker);
//ruta
var objConfigDr = {
map: gMapa
}
var objConfigDs = {
origin:glatlon, //latitud longitud
destination: glatlondestino,
travelMode: google.maps.TravelMode.DRIVING
}
var ds =new google.maps.DirectionsService();//obtener coordenadas
var dr = new google.maps.DirectionsRenderer(objConfigDr);//traduce ruta
ds.route(objConfigDs, fnRutear);
function fnRutear(resultados,status) {
//línea entre el punto a y b
if (status == 'OK') {
dr.setDirections(resultados);
} else {
alert('Error' + status);
}
}
}
</script>

```

**Figura 114.** Método de trazado de ruta en la capa de interfaz  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Ingresar Retiro

El ingreso de retiros presenta una vista de un listado de retiros ingresados que se encuentren pendientes en ser tratados por el personal técnico, la información presentada trata de ayudar al personal de interés a tener una mejor organización de los retiros por atender, en dicha vista presenta la posibilidad de realizar un nuevo ingreso.

**Ingreso de Retiro**

LISTADO DE INGRESOS 👤

Id Ingreso	Cliente	Personal Retiro	Fecha de Ingreso	Observación
4	Valeria Quispe	Fernando Naula	26/06/2022	Completo sin novedades

[Agregar Ingreso](#)

**Figura 115.** Vista de listado de ingresos de retiros

**Elaborado por:** Valeria Quispe

Para realizar el registro de un retiro el personal de entrega y recepción será el encargado de coordinar el ingreso con el usuario a cargo, en ello se seleccionará el retiro a registrar y de ser el caso podrá digitar una observación positiva o negativa conforme al estado de la recolección de los dispositivos.

**Ingreso de Retiro**  
**Registrar Ingreso**

---

DATOS DE LA ASIGNACIÓN DE RETIRO 🕒

Id Retiro:	<input type="text"/>	<input type="button" value="Buscar"/>	Nombre del Cliente:	<input type="text"/>
Dirección Cliente:	<input type="text"/>		Teléfono Cliente:	<input type="text"/>

INFORMACIÓN ADICIONAL DE INGRESO 👤

Id Encargado:	<input type="text"/>	Encargado del Retiro:	<input type="text"/>
Dirección Encargado:	<input type="text"/>	Teléfono Encargado:	<input type="text"/>
Observación:	<input style="width: 100%;" type="text"/>		

---

**Figura 116.** Formulario de ingresar retiro  
**Elaborado por:** Valeria Quispe

Con el fin de ayudar en la gestión de selección de retiro de interés, se presenta un modal emergente con un listado de retiros que se encuentre asignados en ello podrá visualizar la información del cliente, fecha de la asignación del retiro y el nombre de quien fue el responsable de la recolección de los dispositivos tecnológicos.



**Figura 117.** Vista de retiros asignados-Pendientes  
**Elaborado por:** Valeria Quispe

### **Codificación**

Con el fin de llevar a cabo las funciones descritas se expone los métodos como el encargado de recuperar el listado de los retiros ingresados llamado `CargarListadoIngreso()`.

```

private void CargarListadoIngreso()
{
    int estado = 1;
    List<EIngresoRetiro> ListadoIngreso = new List<EIngresoRetiro>();
    ListadoIngreso = NIngresoRetiro.ObtenerListadoIngresosxEstado(estado);
    if (ListadoIngreso == null)
    {
    }
    else
    {
        List<EVistaIngresarRetiro> listaingreso = new List<EVistaIngresarRetiro>();
        foreach (var item in ListadoIngreso)
        {
            EVistaIngresarRetiro vista = new EVistaIngresarRetiro();
            EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxId(item.IdAsignacionRetiro);
            ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);
            EUsuario userCli = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
            EPersona perCli = NPersona.DevolverPersonaxId(userCli.IdPersonaPertenece);

            EUsuario userPer = NUsuario.DevolverUsuarioxId(asignacion.IdUsuarioAsignacion);
            EPersona perPer = NPersona.DevolverPersonaxId(userPer.IdPersonaPertenece);
            vista.IdIngresoRetiro = item.IdIngresoRetiro;
            vista.Cliente = perCli.NombrePersona + " " + perCli.ApellidoPersona;
            vista.PersonalRetiro = perPer.NombrePersona + " " + perPer.ApellidoPersona;
            vista.ObservacionIngreso = item.ObservacionIngreso;
            DateTime fecha = Convert.ToDateTime(item.FechaIngreso);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaIngreso = fecha1;
            listaingreso.Add(vista);
        }
        GridViewAsignaciones.DataSource = listaingreso;
        GridViewAsignaciones.DataBind();
    }
}

```

**Figura 118.** Método cargar listado de ingresos no completados en la capa de datos  
**Elaborado por:** Valeria Quispe

A su vez se presenta el método encargado de presentar el listado de asignaciones de retiros pendientes denominado `CargarListadoAsignacionRetiro()`, con el fin de agilizar el proceso llevado a cabo por el usuario de interés. Cabe destacar que en efecto existe la reutilización de método por ello a su vez se expone lo más importantes en la objeción de cumplir con cada función.

```

private void CargarListadoAsignacionRetiro()
{
    int estado = 1;
    List<EAsignarRetiro> listaAsignaciones = new List<EAsignarRetiro>();
    listaAsignaciones = MAsignarRetiro.ObtenerAsignacionesxEstado(estado);
    List<EVistaAsignarRetiro> listaVista = new List<EVistaAsignarRetiro>();
    if (listaAsignaciones == null)
    {
    }
    else
    {
        foreach (var item in listaAsignaciones)
        {
            ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(item.IdRetiroAsignacion);
            EUsuario cliente = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
            EPersona perCliente = NPersona.DevolverPersonaxId(cliente.IdPersonaPertenece);

            EUsuario responsable = NUsuario.DevolverUsuarioxId(item.IdUsuarioAsignacion);
            EPersona perResponsable = NPersona.DevolverPersonaxId(responsable.IdPersonaPertenece);
            EVistaAsignarRetiro vista = new EVistaAsignarRetiro();

            vista.IdAsignacionRetiro = item.IdAsignacionRetiro;
            vista.Cliente = perCliente.NombrePersona + " " + perCliente.ApellidoPersona;
            vista.DireccionCliente = perCliente.DireccionPersona;
            DateTime fecha = Convert.ToDateTime(item.FechaAsignacion);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaAsignacion = fecha1;
            vista.ResponsableAsignacion = perResponsable.NombrePersona + " " + perResponsable.ApellidoPersona;
            listaVista.Add(vista);
        }
        GridViewRetiros.DataSource = listaVista;
        GridViewRetiros.DataBind();
    }
}

```

**Figura 119.** Método de cargar listado de asignaciones de retiro en la capa de interfaz

**Elaborado por:** Valeria Quispe

Consecuentemente se presenta el método encargado de llevar a cabo el registro del ingreso del retiro, en el mismo método se destaca que se realiza el ingreso del nuevo estado del retiro así con la actualización del estado de la asignación del retiro.



```

private void GuardarIngresoRetiro()
{
    //Guardar Ingreso
    EIngresoRetiro ingreso = new EIngresoRetiro();
    EUsuario user = (EUsuario)Session["Usuario"];
    ingreso.IdIngresoRetiro = 0;
    ingreso.EstadoIngreso = 1;
    ingreso.FechaIngreso = DateTime.Now.ToShortDateString();
    ingreso.IdAsignacionRetiro = Convert.ToInt32(LabelIdAsignacionRetiro.Text);
    ingreso.IdUsuarioIngreso = user.IdUsuario;
    ingreso.ObservacionIngreso = DevolverObservacion();

    ingreso = NIngresoRetiro.GuardarIngresoRetiro(ingreso);

    if (ingreso == null)
    {
    }
    else
    {
        EAsignarRetiro asignacionretiro = NAsignarRetiro.ObtenerAsignacionxId(ingreso.IdAsignacionRetiro);
        //Actualizar Asignacion Retiro a completo
        int estadoAsignacionRetiro = 2;
        EAsignarRetiro asignacion = NAsignarRetiro.ActualizarEstadoAsignacionRetiro(estadoAsignacionRetiro, asignacionretiro.IdAsignacionRetiro);
        //Guardar Estado de Retiro
        EEstadoRetiro estado = new EEstadoRetiro();
        estado.IdEstadoRetiro = 0;
        estado.IdRetiroPertenece = asignacionretiro.IdRetiroAsignacion;
        estado.FechaEstado = ingreso.FechaIngreso;
        estado.EstadoRetiro = "Ingresado";
        estado.IdUsuarioResponsable = ingreso.IdUsuarioIngreso;
        estado.NumEstadoRetiro = 3;
        estado = NEstadoRetiro.GuardarEstado(estado);
        //Actualizar Estado de Retiro
        int estadoRetiro = 3;
        ECabeceraRetiro retiro = new ECabeceraRetiro();
        retiro = NCabeceraRetiro.ActualizarEstadoRetiro(estadoRetiro, asignacionretiro.IdRetiroAsignacion);
        CargarListadoIngreso();
        limpiarCampos();
        MultiView1.ActiveViewIndex = 1;
        LabelMensajeGrid.Text = "Registro realizado correctamente";
    }
}

```

**Figura 120.** Método de registro de ingreso de retiro en la capa de interfaz  
**Elaborado por:** Valeria Quispe

Se presenta el método de la capa de datos que cumple con la finalidad de recibir la información recuperada de la interacción del interfaz y así ser el responsable de poder realizar el registro en el servidor de base de datos, todo esto a través de sentencias SQL.

```

public static EIngresoRetiro GuardarIngresoRetiro(EIngresoRetiro ingreso)
{
    try
    {
        SqlConnection connection = new SqlConnection(Settings1.Default.Conexion);
        connection.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = connection;
        cmd.CommandType = System.Data.CommandType.Text;
        cmd.CommandText = @"INSERT INTO [dbo].[IngresoRetiro]
([IdAsignacionRetiro]
,[IdUsuarioIngreso]
,[FechaIngreso]
,[ObservacionIngreso]
,[EstadoIngreso])
VALUES
(@AsignacionRetiro,@UsuarioIngreso,@Fecha,@Observacion,@Estado);SELECT SCOPE_IDENTITY()";
        cmd.Parameters.AddWithValue("@AsignacionRetiro", ingreso.IdAsignacionRetiro);
        cmd.Parameters.AddWithValue("@UsuarioIngreso", ingreso.IdUsuarioIngreso);
        cmd.Parameters.AddWithValue("@Fecha", ingreso.FechaIngreso);
        cmd.Parameters.AddWithValue("@Observacion", ingreso.ObservacionIngreso);
        cmd.Parameters.AddWithValue("@Estado", ingreso.EstadoIngreso);

        var idIngresoRetiro = Convert.ToInt32(cmd.ExecuteScalar());
        ingreso.IdIngresoRetiro = idIngresoRetiro;
        connection.Close();
        return ingreso;
    }
    catch (Exception)
    {
        return null;
    }
}

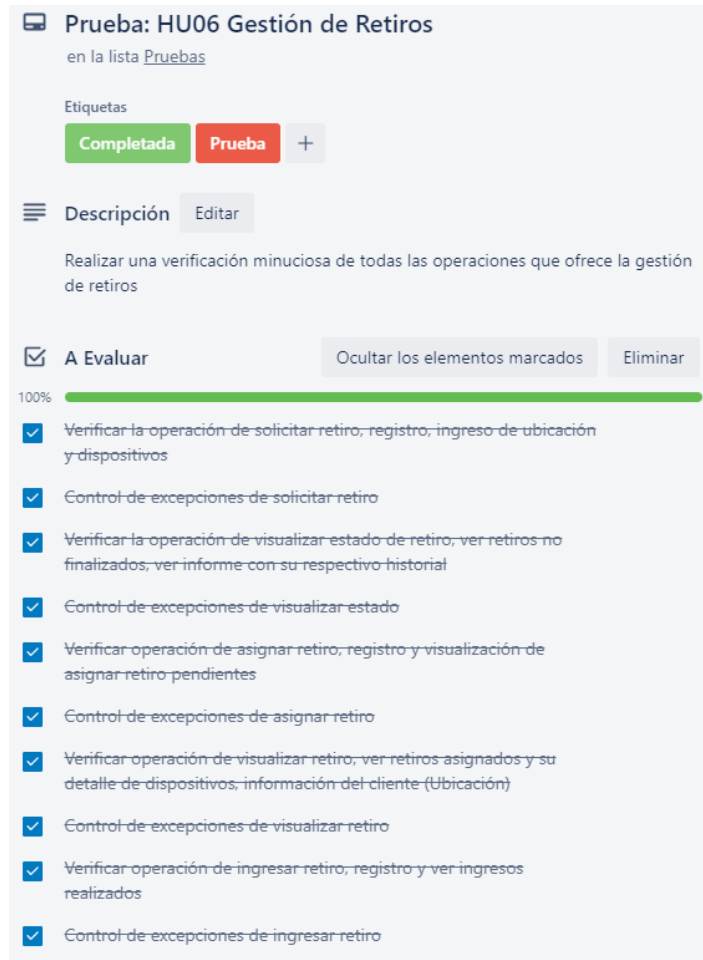
```

**Figura 121.** Método para guardar ingreso de retiro  
**Elaborado por:** Valeria Quispe

**Tarea completada**



**Figura 122.** Tareas y Sub-Tareas de Gestión de Retiros  
**Elaborado por:** Valeria Quispe



**Figura 123.** Pruebas de la Gestión de Retiros  
**Elaborado por:** Valeria Quispe

### **H07 Menú de retiros**

El menú de retiros se enfocada en el desarrollo de la correcta navegación entre las funciones que componen la gestión de retiros, estos habilitándose según el perfil de ingreso del usuario.



**Figura 124.** Menú de Retiros  
**Elaborado por:** Valeria Quispe

### Prueba

La finalidad de la prueba con respecto al menú de retiros, es verificar la correcta habilitación de opciones de la gestión de retiros conforme al perfil de ingreso del usuario, entre los cuales se puede definir como: secretaria, cliente, gerente, personal de entrega y recepción.



**Figura 125.** Pruebas del Menú de Retiros  
**Elaborado por:** Valeria Quispe

### **Evidencia**

En las opciones del menú de retiro se presenta las actividades que puede realizar los diferentes usuarios como lo son, solicitar, ver estado, ver asignación de retiro, asignar retiro e ingresar retiro.



**Figura 126.** Opciones del menú de retiro  
**Elaborado por:** Valeria Quispe

### Codificación

El desarrollo netamente se encuentra enfocado en la realización del menú en la capa de interfaz, donde se define las referencias a las diferentes páginas que componen la gestión de retiros.

```

<div class="sb-nav-link-icon"><i class="fas fa-computer"></i></div>
Retiro
<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
</a>
<div class="collapse" id="collapseLayouts" aria-labelledby="headingOne" data-bs-parent="#sidenavAccordion">
  <nav class="sb-sidenav-menu-nested nav">
    <a class="nav-link" href="ISolicitarRetiro.aspx">Solicitar Retiro</a>
    <a class="nav-link" href="IEstadoRetiros.aspx">Estado de Retiros</a>
    <a class="nav-link" href="IASignarRetiro.aspx">Asignar Retiro</a>
    <a class="nav-link" href="IVerAsignacionRetiro.aspx">Ver Asignación</a>
    <a class="nav-link" href="IIngresarRetiro.aspx">Ingresar Retiro</a>
  </nav>
</div>

```

**Figura 127.** Redirección de formularios de gestión de retiros  
**Elaborado por:** Valeria Quispe

Consecuentemente se presenta la activación del menú de retiros, mismo que es llevado a cabo en la capa de interfaz.

```

<nav class="sb-sidenav-menu-nested nav">
  <div id="opRetiroCliente">
    <a class="nav-link" href="ISolicitarRetiro.aspx">Solicitar Retiro</a>
    <a class="nav-link" href="IEstadoRetiros.aspx">Estado de Retiros</a>
  </div>
  <div id="opRetiroSecretaria">
    <a class="nav-link" href="IAsignarRetiro.aspx">Asignar Retiro</a>
    <a class="nav-link" href="IIngresarRetiro.aspx">Ingresar Retiro</a>
  </div>
  <div id="opRetiroPersonalER">
    <a class="nav-link" href="IVerAsignacionRetiro.aspx">Ver Asignación</a>
  </div>
</nav>

```

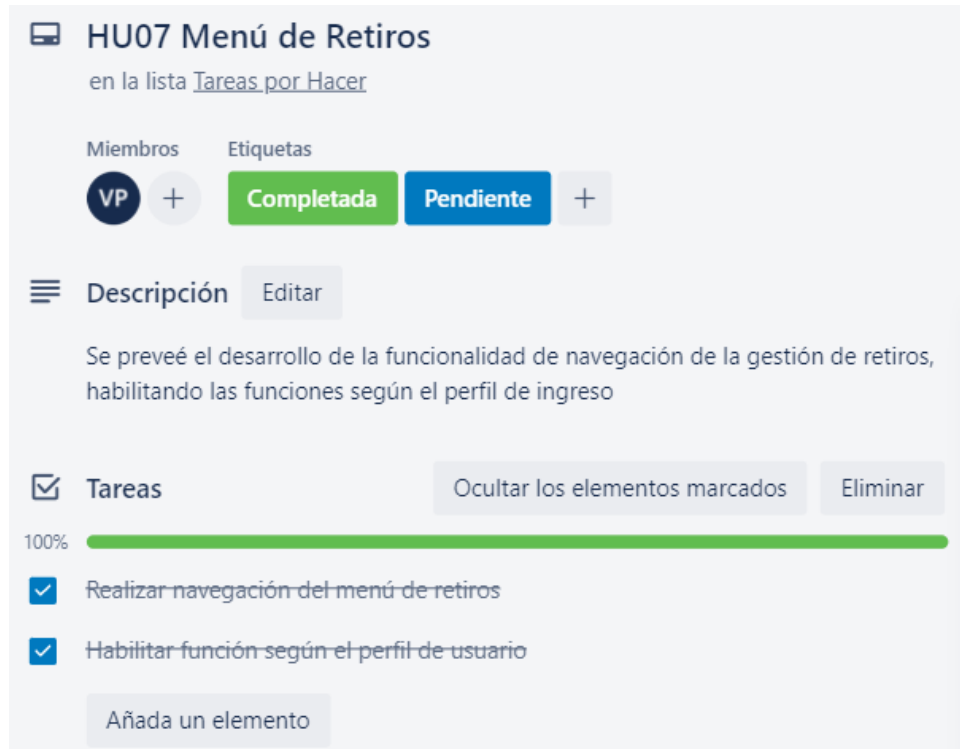
**Figura 128.** Habilitación de operaciones de gestión de retiros por perfil  
 Elaborado por: Valeria Quispe

### Tarea Completada



**Figura 129.** Menú de Retiros completada  
 Elaborado por: Valeria Quispe





**Figura 130.** Menú de Retiros completada  
**Elaborado por:** Valeria Quispe

### **H08 Gestión de diagnósticos**

La gestión de diagnósticos se enfoca en el desarrollo de las funciones que ayuden a manejar dicha actividad, en la cual se estima las funciones de, seleccionar ingreso, ingresar diagnóstico, aceptar diagnóstico y ver diagnósticos.

The screenshot shows a project management interface for a task named "HU08 Gestión de Diagnósticos". At the top, it indicates the task is in the "Tareas por Hacer" list. Below this, there are sections for "Miembros" (Members) and "Etiquetas" (Tags). The "Miembros" section shows a member with the initials "VP" and a plus sign to add more. The "Etiquetas" section shows a blue tag labeled "Pendiente" (Pending) with a plus sign to add more. A "Descripción" (Description) section includes a menu icon, the title "Descripción", and an "Editar" (Edit) button. The description text reads: "Se estima el desarrollo de la gestión de diagnósticos que es llevado a cabo bajo el desarrollo de distintas funciones, con el fin de cumplir cierto objetivo". Below the description is a "Tareas" (Tasks) section with a checkmark icon, the title "Tareas", and an "Eliminar" (Delete) button. A progress bar shows 0% completion. The task list includes four items, each with an unchecked checkbox: "Desarrollar apartado Seleccionar Ingreso", "Desarrollar apartado Ingresar Diagnóstico", "Desarrollar apartado Aceptar Diagnóstico", and "Desarrollar apartado Ver Diagnósticos". At the bottom of the task list is a button labeled "Añada un elemento" (Add an element).

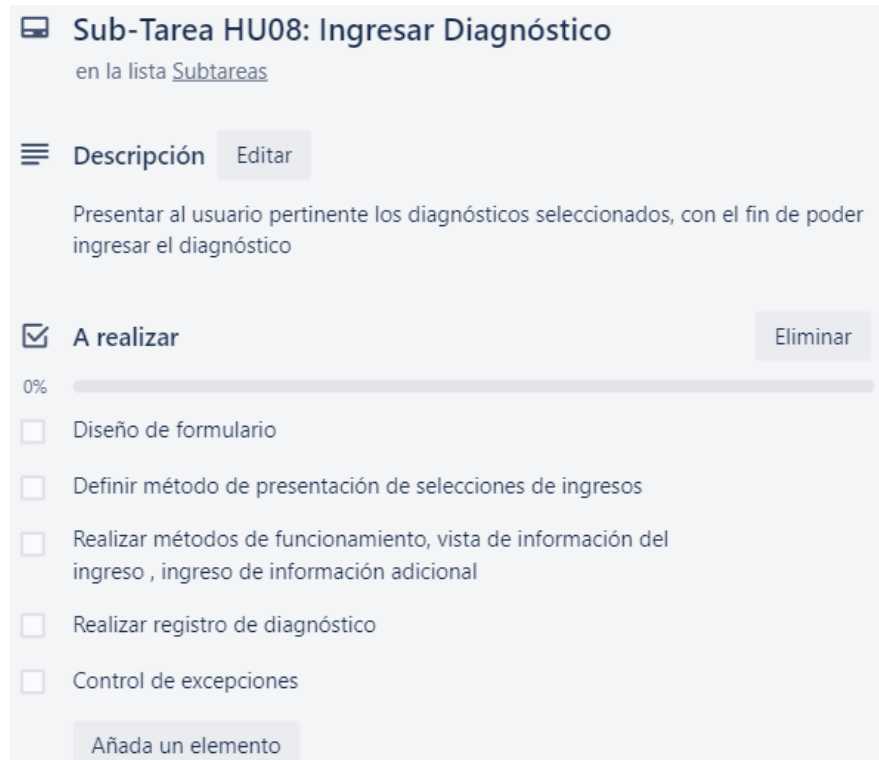
**Figura 131.** Gestión de Diagnósticos  
**Elaborado por:** Valeria Quispe

Para manejar de forma organizada el desarrollo de la gestión de diagnósticos se divide en tareas que ayudarán a agilizar la realización de los mismos, empezando con la subtarea Seleccionar ingreso encargado de poder visualizar los ingresos de retiros en un estado pendiente y ser seleccionado para su previo diagnóstico esto a cargo del personal técnico



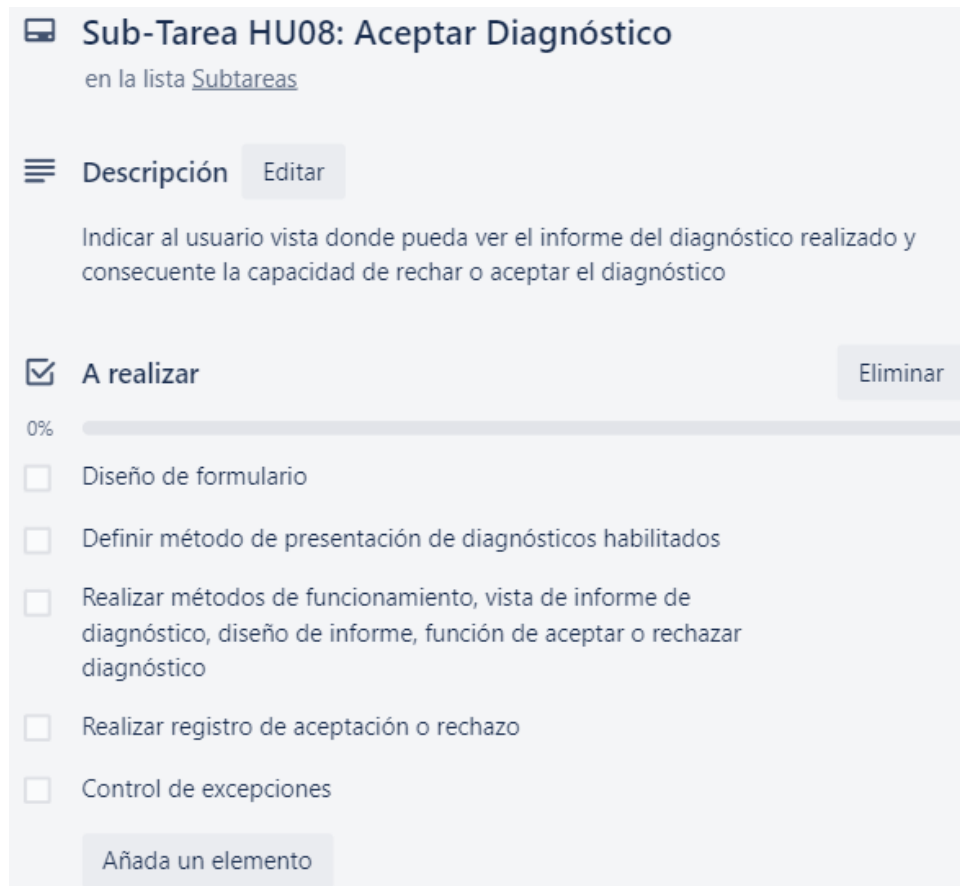
**Figura 132.** Sub-Tarea de la Historia de Usuario HU08 - Seleccionar Ingreso  
**Elaborado por:** Valeria Quispe

La subtarea Ingresar Diagnóstico da la posibilidad en que el usuario encargado pueda hacer el ingreso de la información de la revisión de los dispositivos tecnológico, dando un detallado del fallo, el trabajo de realización y el costo de mismo todo a favor de que el cliente pueda tener un panorama claro del diagnóstico.



**Figura 133.** Sub-Tarea de la Historia de Usuario HU08 - Ingresar Diagnóstico  
**Elaborado por:** Valeria Quispe

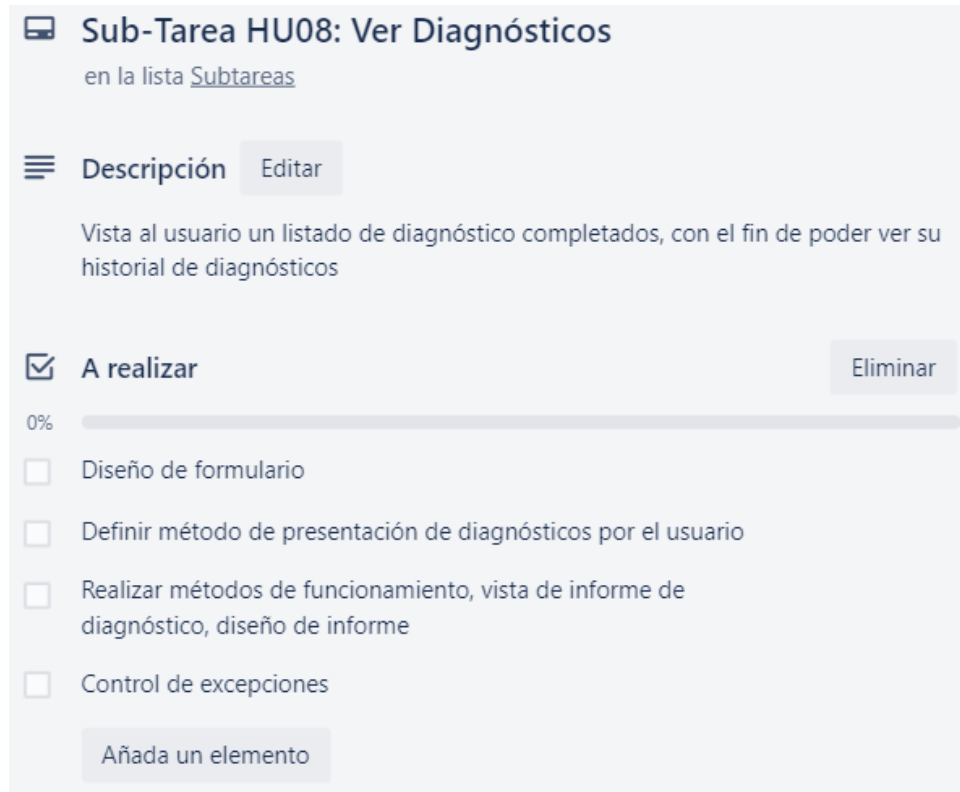
La subtarea Aceptar Diagnóstico permite al cliente visualizar un informe con el diagnóstico realizado por el personal técnico de la empresa, a su vez podrá tener una noción clara del problema en que presenta los dispositivos tecnológicos de esta forma podrá aceptar o rechazar. En caso de que sea positivo se empezará con la reparación o mantenimiento mientras que fuese negativo se procede a coordinar el entrego de los dispositivos.



**Figura 134.** Sub-Tarea de la Historia de Usuario HU08 - Aceptar Diagnóstico

**Elaborado por:** Valeria Quispe

La subtarea Ver Diagnóstico da la posibilidad al usuario que pueda ver un listado de todos los diagnósticos realizados o que se encuentran a su favor con el fin de poder obtener un historial de todos servicios adquiridos.



**Figura 135.** Sub-Tarea de la Historia de Usuario HU08 - Ver Diagnósticos  
**Elaborado por:** Valeria Quispe

### **Prueba**

La finalidad de la prueba trata de llevar una correcta funcionalidad con respecto a la gestión de diagnósticos, evaluando así las subtareas que ayudan a cumplir con el desarrollo de la historia de usuario.



**Figura 136.** Prueba de la Gestión de Diagnósticos  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Seleccionar Ingreso

La selección de ingreso permite visualizar un listado de ingreso de retiros pendientes que no cuentan con un previo diagnóstico, esto a favor de organizar la gestión de diagnósticos indicando consigo la información del cliente como nombre, fecha de ingreso y observación del ingreso.

**Selección de Ingreso**

LISTADO DE INGRESOS DISPONIBLES 📄

Id Ingreso	Cliente	Fecha de Ingreso	Observación	Ver Ingreso
4	Valeria Quispe	26/06/2022	Completo sin novedades	

**Figura 137.** Vista de ingresos disponibles sin diagnóstico  
**Elaborado por:** Valeria Quispe

Para llevar a cabo el registro de la selección de ingreso se presenta un formulario al personal técnico, donde podrá observar información previa del ingreso y de los dispositivos que necesitan de una revisión. Con ello el usuario podrá estimar un tiempo para la realización del diagnóstico antes de registrarlo.

**Selección de Ingreso**  
**Visualización de Retiro**

---

INFORMACIÓN DEL RETIRO 📄

Id Retiro:	<input type="text" value="1"/>	Nombre del Cliente:	<input type="text" value="Valeria Quispe"/>
Dirección:	<input type="text" value="Ciudadela cumanda Alrededor"/>	Teléfono:	<input type="text" value="0979398197"/>
Día - Hora Solicitado:	<input type="text" value="Jueves-15:00"/>	Referencia Domicilio:	<input type="text" value="Frente de upc"/>
Tiempo Estimado, Diagnóstico:	<input type="text" value="0"/>		

DETALLE DE DISPOSITIVOS TECNOLÓGICOS 📄

Id Detalle	Nombre	Marca	Modelo	Descripción
1	CPU	DELL	del-09	Ninguna
2	Impresora	EPSON	C.00	Manuales

**Figura 138.** Formulario de registro de selección de ingreso  
**Elaborado por:** Valeria Quispe

### **Codificación**

Conforme al desarrollo de las funciones se realiza el método para obtener el listado de ingresos disponibles detallado a continuación:



```

private void cargarListadoIngresosDisponibles()
{
    List<EIngresoRetiro> listadoIngresos = new List<EIngresoRetiro>();
    listadoIngresos = NIngresoRetiro.ObtenerListadoIngresosxEstado(1);
    List<EVistaIngresarRetiro> listaVista = new List<EVistaIngresarRetiro>();
    if (listadoIngresos==null)
    {
    }
    else
    {
        foreach (var item in listadoIngresos)
        {
            EVistaIngresarRetiro vista = new EVistaIngresarRetiro();

            EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxId(item.IdAsignacionRetiro);
            ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);
            EUsuario userCli = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
            EPersona perCli = NPersona.DevolverPersonaxId(userCli.IdPersonaPertenece);
            vista.IdIngresoRetiro = item.IdIngresoRetiro;
            DateTime fecha = Convert.ToDateTime(item.FechaIngreso);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaIngreso = fecha1;
            vista.ObservacionIngreso = item.ObservacionIngreso;
            vista.Cliente = perCli.NombrePersona + " " + perCli.ApellidoPersona;
            listaVista.Add(vista);
        }
        GridViewIngresos.DataSource=listaVista;
        GridViewIngresos.DataBind();
    }
}

```

**Figura 139.** Método de recuperación de ingresos disponibles en la capa de interfaz  
**Elaborado por:** Valeria Quispe

En la figura siguiente se detalla el método en cargado de realizar la recuperación de los ingresos que se encuentran con un estado pendiente, mismo de la capa de datos que interactúa con el servidor de base de datos a través de sentencias SQL.

```

public static List<EIngresoRetiro> ObtenerListadoIngresosxEstado(int estado)
{
    try
    {
        List<EIngresoRetiro> listado = new List<EIngresoRetiro>();
        SqlConnection conexion = new SqlConnection(Settings1.Default.Conexion);
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conexion;
        conexion.Open();
        cmd.CommandText = @"SELECT [IdIngresoRetiro]
        ,[IdAsignacionRetiro],[IdUsuarioIngreso]
        ,[FechaIngreso],[ObservacionIngreso]
        ,[EstadoIngreso] FROM [dbo].[IngresoRetiro]
        where EstadoIngreso=" + estado + """;
        cmd.CommandType = System.Data.CommandType.Text;
        using (var dr = cmd.ExecuteReader())
        {
            while (dr.Read())// mientras se pueda leer el datareader
            {
                EIngresoRetiro detalle = new EIngresoRetiro();
                detalle.IdIngresoRetiro = Convert.ToInt32(dr["IdIngresoRetiro"].ToString());
                detalle.IdAsignacionRetiro = Convert.ToInt32(dr["IdAsignacionRetiro"].ToString());
                detalle.IdUsuarioIngreso = Convert.ToInt32(dr["IdUsuarioIngreso"].ToString());
                detalle.FechaIngreso = (dr["FechaIngreso"].ToString());
                detalle.ObservacionIngreso = (dr["ObservacionIngreso"].ToString());
                detalle.EstadoIngreso = Convert.ToInt32(dr["EstadoIngreso"].ToString());
                listado.Add(detalle);
            }
        }
        conexion.Close();
        return listado;
    }catch (Exception)
    {
        return null;
    }
}

```

**Figura 140.** Método de recuperar ingresos disponibles en la capa de datos

**Elaborado por:** Valeria Quispe

Consecuentemente se desataca el método de recuperación de información del ingreso, en el cual hace una recolección de diferentes entidades como datos de asignación, ingreso, retiro y datos personales del cliente.

```

private void cargarRetiro(int id)
{
    EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroxID(id);
    EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxId(ingresoRetiro.IdAsignacionRetiro);
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);

    List<EDetalleRetiro> listadoArticulos = NCabeceraRetiro.ObtenerDetalleRetiroxID(retiro.IdRetiro);
    EUsuario usuario = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
    EPersona cliente = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);

    TextBoxIdRetiro.Text = retiro.IdRetiro.ToString();
    TextBoxDireccion.Text = cliente.DireccionPersona;
    TextBoxDiaHora.Text = retiro.DiaRetiro + "-" + retiro.HoraRetiro;
    TextBoxNombreCliente.Text = cliente.NombrePersona + " " + cliente.ApellidoPersona;
    TextBoxTelefono.Text = cliente.TelefonoPersona;
    TextBoxReferenciaDomi.Text = retiro.ReferenciaDomicilio;
    LabelIdRetiro.Text = (ingresoRetiro.IdIngresoRetiro).ToString();

    GridViewDispositivos.DataSource = listadoArticulos;
    GridViewDispositivos.DataBind();
}

```

**Figura 141.** Método de recuperar información del ingreso en la capa de interfaz

**Elaborado por:** Valeria Quispe

Y previo al realizar el registro de la selección se hace el control del tiempo estimado, validando así las posibles excepciones a futuro, en el método siguiente se detalla la forma de control.

```
protected void ButtonSeleccionarIngreso_Click(object sender, EventArgs e)
{
    if (Convert.ToInt32(TextBoxDiasEstimados.Text) <= 0)
    {
        LabelMensajeRegistro.Text = "Tiempo estimado no puede ser menor a 0";
    }
    else
    {
        GuardarSeleccionIngreso();
    }
}
```

**Figura 142.** Método de control de tiempo estimado selección de ingreso

**Elaborado por:** Valeria Quispe

Y una vez finalizado el proceso anterior se hace el llamado al método de registro de selección, encargado de recuperar la información de los componentes de interfaz para luego enviarlo por medio de entidades con el fin de cumplir el registro en el servidor de base de datos. En el método se destaca el registro del nuevo estado del retiro y consigo la actualización del nuevo estado esto para manejar de forma ordenada las operaciones por la cual pasa un retiro.

```

private void GuardarSeleccionIngreso()
{
    ESeleccionIngreso seleccion = new ESeleccionIngreso();
    EUsuario user= (EUsuario)Session["Usuario"];
    seleccion.IdSeleccionIngreso = 0;
    seleccion.IdIngresoRetiro =Convert.ToInt32(LabelIdRetiro.Text);
    seleccion.FechaSeleccion = DateTime.Now.ToShortDateString();
    seleccion.TiempoEstimado = Convert.ToInt32(TextBoxDiasEstimados.Text);
    seleccion.IdUsuarioSeleccion = user.IdUsuario;
    seleccion.EstadoSeleccion = 1;
    seleccion = NSeleccionIngreso.GuardarSeleccionIngreso(seleccion);
    if (seleccion == null)
    {
    }
    else
    {
        //Guardar Nuevo Estado
        EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroxID(seleccion.IdIngresoRetiro);
        EAsignarRetiro asignacionRetiro = NAsignarRetiro.ObtenerAsignacionxId(ingresoRetiro.IdAsignacionRetiro);
        EEstadoRetiro estado = new EEstadoRetiro();
        estado.IdEstadoRetiro = 0;
        estado.IdRetiroPertenece = asignacionRetiro.IdRetiroAsignacion;
        estado.FechaEstado = seleccion.FechaSeleccion;
        estado.EstadoRetiro = "Revision";
        estado.IdUsuarioResponsable = seleccion.IdUsuarioSeleccion;
        estado.NumEstadoRetiro = 4;
        estado = NEstadoRetiro.GuardarEstado(estado);
        //Actualizar Estado de Ingreso
        int estadoIngreso = 2;
        EIngresoRetiro ingreso = new EIngresoRetiro();
        ingreso = NIngresoRetiro.ActualizarEstadoIngreso(estadoIngreso, ingresoRetiro.IdIngresoRetiro);

        //Actualizar Estado Retiro
        int estadoRetiro = 4;
        ECabeceraRetiro retiro = new ECabeceraRetiro();
        retiro = NCabeceraRetiro.ActualizarEstadoRetiro(estadoRetiro, asignacionRetiro.IdRetiroAsignacion);

        cargarListadoIngresosDisponibles();
        limpiarCampos();
        MultiView1.ActiveViewIndex = 1;
        LabelMensajeGrid.Text = "Registro realizado correctamente";
    }
}

```


**Figura 143.** Método de guardar selección de ingreso en la capa de interfaz  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Ingresar Diagnóstico

El proceso de ingresar diagnóstico presenta al usuario un listado de selecciones de ingresos que se encuentran pendiente, con el fin de facilitar el proceso al técnico en el mismo podrá constatar de información del cliente, fecha de selección y tiempo estimado.

#### Visualización de Ingresos Seleccionados

LISTADO DE INGRESOS POR DIAGNOSTICAR				
Id Selección	Cliente	Fecha de Selección	Tiempo Estimado	Agregar Diagnóstico
2	Valeria Quispe	26/06/2022	2 Días	

**Figura 144.** Vista de ingresos seleccionados por personal técnico  
**Elaborado por:** Valeria Quispe

Consecuente para realizar el registro del diagnóstico se visualiza un formulario donde se presenta una breve información de la selección y a su vez pone un apartado para la selección del dispositivo a ingresar el diagnóstico esto hasta completar una lista de todos los dispositivos con su respectivo análisis. De forma en que se apoye aceptación o rechazo se especifica el precio del arreglo o mantenimiento de todos los dispositivos tecnológicos.

**Visualización de Ingresos Seleccionados**  
**Ingreso de Diagnóstico**

INFORMACIÓN DEL RETIRO:

Id Retiro: <input type="text" value="1"/>	Nombre del Cliente: <input type="text" value="Valeria Quispe"/>
Dirección: <input type="text" value="Ciudadela cumanda Alrededor"/>	Teléfono: <input type="text" value="0979398197"/>
Día - Hora Solicitado: <input type="text" value="Jueves 15:00"/>	Referencia Domicilio: <input type="text" value="Frente de upc"/>

DISPOSITIVOS

Id Detalle: <input type="text"/> <span style="float: right; font-size: x-small;">Buscar</span>	Nombre Dispositivo: <input type="text"/>
Marca: <input type="text"/>	Modelo: <input type="text"/>

INFORMACIÓN DE DIAGNÓSTICO

Descripción: <input type="text"/>	
Tipo de Servicio: <input type="text" value="Seleccione"/>	Valor (\$): <input type="text"/>

Agregar Diagnóstico

**Mensaje : Diagnóstico agregado correctamente**

Quitar	Detalle	Descripción	Tipo	Precio
	1	Necesita actualización de software	Mantenimiento	12
				<b>SUB-TOTAL:</b>
				12
				<b>IVA:</b>
				1,44
				<b>TOTAL:</b>
				13,44

Cancelar
Guardar

**Figura 145.** Formulario de ingreso de diagnóstico  
**Elaborado por:** Valeria Quispe

La ventana emergente ayuda al usuario a realizar el diagnóstico para que no quede ningún dispositivo sin su previa revisión y así generar el registro sin ningún inconveniente.



**Figura 146.** Vista emergente de dispositivos a diagnosticar  
Elaborado por: Valeria Quispe

### Codificación

Para llevar a cabo la subtask se destaca el método encargado de presentar listado de ingresos seleccionados, el mismo presentará la información que corresponde al usuario de interés con el fin de que el personal técnico pueda realizar el ingreso del diagnóstico.

```
private void CargarListadoSeleccionIngreso(int idUsuario)
{
    List<ESeleccionIngreso> listaSeleccionAsignado = new List<ESeleccionIngreso>();
    int estado = 1;
    listaSeleccionAsignado = NSeleccionIngreso.ObtenerSeleccionesxEstadoUsuario(estado, idUsuario);
    if (listaSeleccionAsignado == null)
    {
    }
    else
    {
        List<EVistaSeleccionIngreso> listaVista = new List<EVistaSeleccionIngreso>();
        foreach (var item in listaSeleccionAsignado)
        {
            EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroxID(item.IdIngresoRetiro);
            EAsignarRetiro asignacionRetiro = NAsignarRetiro.ObtenerAsignacionxId(ingresoRetiro.IdAsignacionRetiro);
            ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacionRetiro.IdRetiroAsignacion);
            EUsuario cliente = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
            EPersona perCliente = NPersona.DevolverPersonaxId(cliente.IdPersonaPertenece);
            EVistaSeleccionIngreso vista = new EVistaSeleccionIngreso();
            vista.IdSeleccionIngreso = item.IdSeleccionIngreso;
            DateTime fecha = Convert.ToDateTime(item.FechaSeleccion);
            string fecha1 = fecha.ToString("dd/MM/yyyy");
            vista.FechaSeleccion = fecha1;
            vista.TiempoEstimado = (item.TiempoEstimado).ToString()+" "+ "Días";
            vista.Cliente=perCliente.NombrePersona+" "+perCliente.ApellidoPersona;

            listaVista.Add(vista);
        }
        GridViewSelecciones.DataSource = listaVista;
        GridViewSelecciones.DataBind();
    }
}
```

**Figura 147.** Método de obtener selecciones de ingreso por usuario en la capa de interfaz  
Elaborado por: Valeria Quispe

Para que el personal técnico pueda empezar a realizar el ingreso de diagnóstico, se cumple la selección del ingreso de interés, en el cual se recupera la información de los dispositivos a evaluar e información del cliente.

```
private void EnviarSeleccionRetiro(GridViewCommandEventArgs e)
{
    int index = Convert.ToInt32(e.CommandArgument);
    GridViewRow row = GridViewSelecciones.Rows[index];
    int IdSeleccionIngreso = Convert.ToInt32(row.Cells[0].Text);

    ESeleccionIngreso seleccionIngreso = NSeleccionIngreso.ObtenerSeleccionIngresoById(IdSeleccionIngreso);
    EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroById(seleccionIngreso.IdIngresoRetiro);
    EAsignarRetiro asignacionRetiro = NAsignarRetiro.ObtenerAsignacionById(ingresoRetiro.IdAsignacionRetiro);
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroById(asignacionRetiro.IdRetiroAsignacion);
    EUsuario usuario = NUsuario.DevolverUsuarioById(retiro.IdUsuarioRetiro);
    EPersona persona = NPersona.DevolverPersonaById(usuario.IdPersonaPertenece);

    List<EDetalleRetiro> listaDispositivos = new List<EDetalleRetiro>();
    listaDispositivos = NCabeceraRetiro.ObtenerDetalleRetiroById(retiro.IdRetiro);

    LabelIdRetiro.Text = IdSeleccionIngreso.ToString();
    TextBoxDireccion.Text = persona.DireccionPersona;
    TextBoxIdRetiro.Text = retiro.IdRetiro.ToString();
    TextBoxNombreCliente.Text = persona.NombrePersona + " " + persona.ApellidoPersona;
    TextBoxTelefono.Text = persona.TelefonoPersona;
    TextBoxReferenciaDomi.Text = retiro.ReferenciaDomicilio;
    TextBoxDiaHora.Text = retiro.DiaRetiro + " " + retiro.HoraRetiro;
    Session["ListaDispositivos"] = listaDispositivos;
    GridViewDispositivos.DataSource = listaDispositivos;
    GridViewDispositivos.DataBind();

    MultiView1.ActiveViewIndex = 0;
}
```

**Figura 148.** Método de recuperar información de la selección de ingreso  
Elaborado por: Valeria Quispe

Conjuntamente se presenta el método que cumple la ventana emergente para la revisión individual de cada dispositivo tecnológico. Previo a realizar la revisión de cada dispositivo se emplea el método a llenará una lista con toda la información de la revisión realizada.

```

protected void AgregarDiagnostico()
{
    try
    {
        DataTable dt = new DataTable();
        dt = (DataTable)Session["dt"];
        DataRow[] dr;
        dr = dt.Select("Detalle=" + TextBoxIdDetalleDispositivo.Text);
        if (dr.Length > 0)
        {
            LabelMensaje.ForeColor = System.Drawing.Color.DarkRed;
            LabelMensaje.Text = "<b>Mensaje</b> : Dispositivo ya se encuentra con su respectivo diagnóstico";
            return;
        }
        string descripcion = "", servicio = "", precio = "", id="";
        if (TextBoxDescripcionDiagnostico.Text == "" || TextBoxPrecio.Text==" ||
            DropDownListTipoTrabajo.SelectedIndex <= 0 || Convert.ToInt32(TextBoxPrecio.Text)<0
            || TextBoxIdDetalleDispositivo.Text=="")
        {
            LabelMensaje.ForeColor = System.Drawing.Color.Red;
            LabelMensaje.Text = "<b>Mensaje</b> :Datos incompletos para agregar diagnóstico";
        }
        else{
            id = TextBoxIdDetalleDispositivo.Text;
            precio = TextBoxPrecio.Text;
            descripcion = TextBoxDescripcionDiagnostico.Text;
            servicio = DropDownListTipoTrabajo.SelectedItem.ToString();
            dt.Rows.Add(id, descripcion, servicio, precio);
            SumTotal(dt);
            GridViewDetalle.DataSource = dt;
            GridViewDetalle.DataBind();
            LabelMensaje.ForeColor = System.Drawing.Color.Green;
            LabelMensaje.Text = "<b>Mensaje</b> : Diagnóstico agregado correctamente";
            limpiarCamposDiagnostico();

            Session["dt"] = dt;
        }
    }
    catch (Exception)
    {
    }
}

```

**Figura 149.** Método de agregar revisión de dispositivo  
Elaborado por: Valeria Quispe

En ello también se calcula el valor total del diagnóstico esto para que el cliente pueda confirmar o rechazar la reparación.

```

protected void SumTotal(DataTable dt)
{
    float sum = 0;
    float discount = 0;
    double iva;
    float total = 0;

    foreach (DataRow item in dt.Rows)
    {
        sum += Convert.ToSingle(item["Precio"].ToString());
    }
    total = sum - discount;
    LabelSubTotal.Text = Math.Round(sum, 2).ToString();
    iva = (sum * 0.12);
    LabelIVA.Text = Math.Round(iva, 2).ToString();
    // lblDiscount.Text = discount.ToString();
    LabelTOTAL.Text = Math.Round(total + iva, 2).ToString();
    //lblTotal.Text = (total).ToString();
}

```

**Figura 150.** Método de cálculo de valor total  
Elaborado por: Valeria Quispe



Y para culminar con el registro del diagnóstico se lleva a cabo el desarrollo del método GuardarDiagnostico(), que cumple con ciertas funciones como la validación de la información de registro, registro del diagnóstico, registro de nuevo estado de retiro y actualización de estado de selección.

```
private void GuardarDiagnostico()
{
    int numDispositivos = ((List<EDetalleRetiro>)Session["ListaDispositivos"]).Count;
    int numDispositivosDiagnosticados = ((DataTable)Session["dt"]).Rows.Count;
    if (numDispositivos != numDispositivosDiagnosticados)
    {
        LabelMensaje.Text = "Dispositivos Faltantes por ingresar Diagnóstico"+numDispositivos+"-"+numDispositivosDiagnosticados;
    }
    else
    {
        EUsuario user = (EUsuario)Session["Usuario"];
        //LabelMensaje.Text = "Dispositivos"+numDispositivos+"-"+numDispositivosDiagnosticados;
        DataTable dt = new DataTable();
        dt = (DataTable)Session["dt"];
        EDiagnosticoCabecera diagnostico = new EDiagnosticoCabecera();
        diagnostico.IdDiagnostico = 0;
        diagnostico.IdSeleccionPertenece = Convert.ToInt32(LabelIdRetiro.Text.ToString());
        diagnostico.FechaDiagnostico = DateTime.Now.ToShortDateString();
        diagnostico.IdUsuarioDiagnostico = user.IdUsuario;
        diagnostico.EstadoDiagnostico = 1;
        diagnostico.EstadoDiagnostico = 1;
        diagnostico.AceptacionDiagnostico = 0;
        diagnostico.TotalDiagnostico = float.Parse(LabelTOTAL.Text);
        diagnostico=NDiagnosticoCabecera.GuardarDiagnostico(diagnostico, dt);

        if (diagnostico == null)
        {
            LabelMensaje.Text = "Registro no completado";
        }
        else
        {
            //Guardar Nuevo Estado Retiro

            ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoID(diagnostico.IdSeleccionPertenece);
            EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroID(seleccion.IdIngresoRetiro);
            EAsignarRetiro asignacionRetiro = NAsignarRetiro.ObtenerAsignacionID(ingresoRetiro.IdAsignacionRetiro);

            EEstadoRetiro estado = new EEstadoRetiro();
            estado.IdEstadoRetiro = 0;
            estado.IdRetiroPertenece = asignacionRetiro.IdRetiroAsignacion;
            estado.FechaEstado = diagnostico.FechaDiagnostico;
            estado.EstadoRetiro = "Diagnosticado";
            estado.IdUsuarioResponsable = diagnostico.IdUsuarioDiagnostico;
            estado.NumEstadoRetiro = 5;
            estado = NEstadoRetiro.GuardarEstado(estado);
            //Actualizar estado de selección

            int estadoSeleccion = 2;
            ESeleccionIngreso seleccionA = new ESeleccionIngreso();
            seleccionA = NSeleccionIngreso.ActualizarEstadoSeleccion(estadoSeleccion,seleccion.IdSeleccionIngreso);
            //Actualizar Estado de retiro
            int estadoRetiro = 5;
            ECabeceraRetiro retiro = new ECabeceraRetiro();
            retiro = NCabeceraRetiro.ActualizarEstadoRetiro(estadoRetiro, asignacionRetiro.IdRetiroAsignacion);
            CargarListadoSeleccionIngreso(user.IdUsuario);
            limpiarTodosCampos();
            MultiView1.ActiveViewIndex = 1;
            LabelMensajeGrid.Text = "Diagnóstico Realizado Correctamente";
        }
    }
}
```

**Figura 151.** Método de registro de diagnóstico en la capa de interfaz  
**Elaborado por:** Valeria Quispe

De la misma forma se presenta el método perteneciente a la capa de datos que realiza el registro tanto de la cabecera como el detalle de del diagnóstico.

```

public static EDiagnosticoCabecera GuardarDiagnostico(EDiagnosticoCabecera diagnostico, DataTable dt)
{
    try
    {
        int confirm;
        string query = "INSERT INTO [dbo].[DiagnosticoCabecera] ([IdSeleccionPertenece] +
            ",[FechaDiagnostico],[IdUsuarioDiagnostico]" +
            ",[EstadoDiagnostico],[EstadoEliminacion]" +
            ",[AceptacionDiagnostico],[TotalDiagnostico])VALUES" +
            " (@seleccion," +
            " @fecha," +
            " @usuario," +
            " @estado," +
            " @eliminacion," +
            " @aceptacion," +
            " @total)" +
            " SELECT SCOPE_IDENTITY()";

        string dtquery = "INSERT INTO [dbo].[DiagnosticoDetalle]([IdRetiroDetallePertenece]" +
            ",[DescripcionDiagnostico],[TipoServicio]" +
            ",[Precio],[IdDiagnosticoPertenece])VALUES" +
            " (@idRetiroDetalle," +
            " @descripcion," +
            " @tipo," +
            " @precio," +
            " @diagnostico)";

        using (SqlConnection connection = new SqlConnection((Settings1.Default.Conexion)))
        {
            connection.Open();
            var transaccion = connection.BeginTransaction();

            using (SqlCommand cmd = new SqlCommand(query, connection, transaccion))
            {
                cmd.Parameters.AddWithValue("@seleccion", diagnostico.IdSeleccionPertenece);
                cmd.Parameters.AddWithValue("@fecha", diagnostico.FechaDiagnostico);
                cmd.Parameters.AddWithValue("@usuario", diagnostico.IdUsuarioDiagnostico);

                cmd.Parameters.AddWithValue("@estado", diagnostico.EstadoDiagnostico);
                cmd.Parameters.AddWithValue("@eliminacion", diagnostico.EstadoEliminacion);
                cmd.Parameters.AddWithValue("@aceptacion", diagnostico.AceptacionDiagnostico);
                cmd.Parameters.AddWithValue("@total", diagnostico.TotalDiagnostico);

                diagnostico.IdDiagnostico = Convert.ToInt32(cmd.ExecuteScalar());
                if (diagnostico.IdDiagnostico > 0)
                    confirm = 1;
                else
                    confirm = -1;
            }

            using (SqlCommand cmd = new SqlCommand(dtquery, connection, transaccion))
            {
                foreach (DataRow item in dt.Rows)
                {
                    cmd.Parameters.Clear();
                    cmd.Parameters.AddWithValue("@idRetiroDetalle", item["Detalle"]);
                    cmd.Parameters.AddWithValue("@descripcion", item["Descripcion"]);
                    cmd.Parameters.AddWithValue("@tipo", item["Tipo"]);
                    cmd.Parameters.AddWithValue("@precio", item["Precio"]);
                    cmd.Parameters.AddWithValue("@diagnostico", diagnostico.IdDiagnostico);
                    cmd.ExecuteNonQuery();
                }
            }

            transaccion.Commit();
            connection.Close();
        }

        return diagnostico;
    }
    catch (Exception)
    {
        return null;
    }
}

```

**Figura 152.** Método de registro de diagnóstico en la capa de datos  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Aceptar Diagnóstico

La subtarea de aceptación o rechazo de diagnóstico permite al usuario visualizar todos los diagnósticos pendientes a una respuesta, en ello podrá obtener un informe detallado o tendrá la capacidad de poder aceptar o rechazar conforme a sus posibilidades.

**Visualización de Diagnósticos**

LISTADO DE DIAGNÓSTICOS

Id Diagnóstico	Fecha	Total	Ver Diagnóstico
	2 26/06/2022	24,64	  

**Figura 153.** Vista de listado de diagnóstico perteneciente a un cliente  
**Elaborado por:** Valeria Quispe

El informe del diagnóstico presenta una información detallada del retiro que incluye información personal como: nombres, dirección domiciliaria, dispositivos del retiro, diagnóstico correspondiente a cada dispositivo y el valor de la reparación o mantenimiento.



## EPC-COMPU

RICARDO PÉREZ GERENTE

Dirección: La Quigo-Atocha

Mantenimiento y Reparación de Equipos Electrónicos



### DIAGNÓSTICO DEL RETIRO N°1

#### INFORMACIÓN DEL CLIENTE

<b>Nombre:</b>	Valeria Quispe	<b>RUC / CI:</b>	1805153580
<b>Dirección:</b>	Ciudadela curanda Alrededor	<b>Fecha Solicitada:</b>	29/03/2022
<b>Teléfono:</b>	0979398197	<b>Día y Hora estimada:</b>	Jueves-15:00
<b>Referencia:</b>	Frete de upc	<b>Retiro:</b>	1
<b>Responsable Diagnóstico:</b>	David Cervantesi	<b>Total, Diagnóstico(S):</b>	<b>24,64</b>

#### ARTÍCULOS DEL RETIRO

Nombre	Marca	Modelo	Descripción	Servicio
CPU	DELL	del-09	Ninguna	Mantenimiento
Impresora	EPSON	C.00	Manuales	Mantenimiento

#### DIAGNÓSTICO DE DEPOSITIVOS

Dispositivo	Descripción	Precio
CPU-DELL	Necesita actualización de software	12
Impresora-EPSON	Necesita actualización de software	10
<b>Sub-Total:</b>		<b>22</b>
<b>IVA:</b>		<b>2,64</b>
<b>TOTAL:</b>		<b>24,64</b>

#### DESCRIPCIÓN DE ACEPTACIÓN O RECHAZO

Estado	Descripción
Aceptar	Una vez aceptado el diagnóstico se procederá a realizar la REPARACIÓN O MANTENIMIENTO de los dispositivos, conforme al precio establecido y se podrá visualizar el estado del mismo a través del sistema
Rechazar	Una vez rechazado el diagnóstico se finalizará el proceso, informando al usuario el acercamiento para el retiro de los dispositivos

26/06/2022

Informe emitido por Sistema Web EPC-Compu

**Figura 154.** Informe de diagnóstico  
Elaborado por: Valeria Quispe

Los mensajes de aceptación o rechazo permite tener una idea clara antes de aceptar o rechazar el diagnóstico.



**Figura 155.** Ventanas emergentes de aceptación de diagnóstico  
Elaborado por: Valeria Quispe



**Figura 156.** Ventanas emergentes de rechazo de diagnóstico  
**Elaborado por:** Valeria Quispe

### Codificación

Se presenta el método encargado de presentar el listado de diagnósticos completados y listos para aceptar o rechazar por el cliente.

```
private void CargarListadoDiagnosticosxIdUsuario(int idUsuario)
{
    List<EDiagnosticoCabecera> Listadiagnostico = new List<EDiagnosticoCabecera>();
    int estadoDiagnostico = 1;
    Listadiagnostico = NDiagnosticoCabecera.ObtenerDiagnosticosxIdUsuario(idUsuario, estadoDiagnostico);
    List<EDiagnosticoCabecera> listaLimpia = new List<EDiagnosticoCabecera>();
    foreach (var item in Listadiagnostico)
    {
        EDiagnosticoCabecera diagnostico = new EDiagnosticoCabecera();
        diagnostico.IdDiagnostico = item.IdDiagnostico;
        DateTime fecha = Convert.ToDateTime(item.FechaDiagnostico);
        string fecha1 = fecha.ToString("dd/MM/yyyy");
        diagnostico.FechaDiagnostico = fecha1;
        diagnostico.TotalDiagnostico = item.TotalDiagnostico;
        listaLimpia.Add(diagnostico);
    }

    GridViewDiagnostico.DataSource = listaLimpia;
    GridViewDiagnostico.DataBind();
}
```

**Figura 157.** Método de obtener diagnósticos por usuario en la capa de interfaz

**Elaborado por:** Valeria Quispe

Previo a la aceptación o rechazo el usuario de interés podrá visualizar el informe donde se detalla la información del diagnóstico, con el valor total de la reparación.

```

private void GenerarPDFDiagnostico(GridviewCommandEventArgs e)
{
    int index = Convert.ToInt32(e.CommandArgument);
    GridViewRow row = GridViewDiagnostico.Rows[index];
    int idDiagnostico = Convert.ToInt32(row.Cells[0].Text);
    EDiagnosticoCabecera diagnostico = NDiagnosticoCabecera.ObtenerDiagnosticoID(idDiagnostico);
    ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoID(diagnostico.IdSeleccionPertenece);
    EIngresoRetiro ingreso = NIngresoRetiro.ObtenerIngresoRetiroID(seleccion.IdIngresoRetiro);
    EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionID(ingreso.IdAsignacionRetiro);
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroID(asignacion.IdRetiroAsignacion);
    DateTime fechaSolicitada = Convert.ToDateTime(retiro.FechaSolicitada);
    string fechaSol = fechaSolicitada.ToString("dd/MM/yyyy");
    EUsuario usuario = NUsuario.DevolverUsuarioID(retiro.IdUsuarioRetiro);
    EPersona persona = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);

    EUsuario userDia = NUsuario.DevolverUsuarioID(diagnostico.IdUsuarioDiagnostico);
    EPersona perDia = NPersona.DevolverPersonaxId(userDia.IdPersonaPertenece);
    List<DetalleRetiro> listaDetalleRetiro = NCabeceraRetiro.ObtenerDetalleRetiroID(retiro.IdRetiro);
    string detalleRetiro = "";
    foreach (var item in listaDetalleRetiro)
    {
        detalleRetiro += "<tr><td style='width:18%;>" + item.NombreArticulo + "</td><td style='width:18%;>" + item.MarcaArticulo + "</td>" +
            "<td style='width:18%;>" + item.ModeloArticulo + "</td><td style='width:24%;>" + item.DescripcionArticulo + "</td>" +
            "<td style='width:18%;>" + item.TipoServicio + "</td></tr>";
    }

    List<EDiagnosticoDetalle> listadDetalleDiagnostico = NDiagnosticoCabecera.ObtenerDetalleDiagnosticoID(diagnostico.IdDiagnostico);
    string detalleDiagnostico = "";
    float suma=0;
    double iva = 0;

    foreach (var item in listadDetalleDiagnostico)
    {
        //DateTime fechaH = Convert.ToDateTime(item.);
        //string fechaL = fechaH.ToString("dd/MM/yyyy");
        EDiagnosticoDetalle detalle = NDiagnosticoCabecera.ObtenerDetalleDiagnosticoSoloxIdDetalle(item.IdDiagnosticoDetalle);

        EDetalleRetiro detalleRe = NCabeceraRetiro.ObtenerDetalleSoloxId(detalle.IdRetiroDetallePertenece);

        detalleDiagnostico += "<tr><td style='width:33%;>" + detalleRe.NombreArticulo + "-" + detalleRe.MarcaArticulo + "</td>" +
            "<td style='width:33%;>" + detalle.DescripcionDiagnostico + "</td><td style='width:33%;text-align:right;>" + detalle.Precio + "</td></tr>";
        suma += detalle.Precio;
    }

    iva=(suma*0.12);

    string fecha = DateTime.Now.ToShortDateString();

    string html = "<DOCTYPE html>" +
        "<html><head><meta charset='UTF-8'><title> Diagnóstico del Retiro N°" + retiro.IdRetiro + " </title>" +
        "</head><style>.class_table {border-collapse: separate;border-spacing: 10px;border: 1px solid black;border-radius: 15px;moz-border-radius: 20px;padding: 2px;}</style></head><body><div style='text-align:center;><img alt='Logo' src='https://i.ibb.co/5541ngS/Logo.png' width='100px' height='100px'></div>" +
        "<div style='text-align:center;><h1 style='text-align:center;>EPC-COMPU </h1><p>RICARDO PÉREZ GERENTE</p><p>Dirección: La Quigo-Atocha </p></div>" +
        "<div style='text-align:center;><img alt='Logo-Ambato' src='https://i.ibb.co/m01trrd/Logo-Ambato.jpg' width='100px' height='100px'></div>" +
        "<div style='text-align:center;><h3>DIAGNÓSTICO DEL RETIRO N°" + retiro.IdRetiro + " </h3></div>" +
        "</div></td></tr>" +
        "</table>" +
        "<table style='width: 100%;' class='class_table'>" +
        "<tr><td style='width:2%;></td>" +
        "<td colspan='5'><b> INFORMACIÓN DEL CLIENTE</b></td>" +
        "<td style='width:2%;></td></tr>" +
        "<tr><td style='width:2%;></td><td style='width:20%;><b> Nombre:</b></td>" +
        "<td style='width:28%;>" + persona.NombrePersona + " " + persona.ApellidoPersona + "</td><td style='width:2%;></td>" +
        "<td style='width:20%;><b> RUC / CI:</b></td>" +
        "<td style='width:28%;>" + retiro.CedoRuc + "</td><td style='width:2%;></td></tr>" +
        "<tr><td style='width:2%;></td><td style='width:20%;><b> Dirección:</b></td>" +
        "<td style='width:28%;>" + persona.DireccionPersona + " </td><td style='width:2%;></td>" +
        "<td style='width:20%;><b> Fecha Solicitada:</b></td><td style='width:28%;>" + fechaSol + "</td>" +
        "<td style='width:2%;></td></tr><tr><td style='width:2%;></td><td style='width:20%;><b> Teléfono:</b></td>" +
        "<td style='width:28%;>" + persona.TelefonoPersona + " </td><td style='width:2%;></td><td style='width:20%;><b> Día y Hora estimada:</b></td>" +
        "<td style='width:28%;>" + retiro.DiaRetiro + " " + retiro.HoraRetiro + " </td><td style='width:2%;></td></tr>" +
        "<tr><td style='width:2%;></td><td style='width:20%;><b> Referencia:</b></td><td style='width:28%;>" + retiro.ReferenciaDomicilio + "</td>" +
        "<td style='width:2%;></td><td style='width:20%;><b> Retiro:</b></td><td style='width:28%;>" + retiro.IdRetiro + "</td>" +
        "<td style='width:2%;></td></tr>" +
        "<tr><td style='width:2%;></td><td style='width:20%;><b> Responsable Diagnóstico:</b></td>" +
        "<td style='width:28%;>" + perDia.NombrePersona + " " + perDia.ApellidoPersona + "</td><td style='width:2%;></td>" +
        "<td style='width:20%;><b> Total, Diagnóstico($):</b></td>" +
        "<td style='width:28%; color:red;>" + suma + "</td><td style='width:2%;></td></tr>" +
        "</table>" +
        "<br><table style='width: 100%;' class='class_table'><tr><td style='width:2%;></td><td colspan='5'><b>ARTÍCULOS DEL RETIRO</b></td><td style='width:2%;></td>" +
        "<tr><td style='width:2%;></td><td colspan='5'><table style='width: 100%; border='1'><tr><td style='background-color: gainsboro;>" +
        "<td style='width:18%;><b>Nombre</b></td>" +

```

```

"td style='width:18%;<b>Marca</b>'/td" +
"td style='width:18%;<b>Modelo</b>'/td" +
"td style='width:24%;<b>Descripción</b>'/td" +
"td style='width:18%;<b>Servicio</b>'/td"></tr></table></table> " +
"<br><table style='width: 100%; class='clase_table'><tr><td style='width:2%;</td>" +
"td colspan='3'><b>DIAGNÓSTICO DE DEPOSITIVOS</b></td><td style='width:2%;</td></tr><tr>" +
"td style='width:2%;</td><td colspan='3'><table style='width: 100%; border='1'>" +
"tr style='background-color: gainsboro;'><td style='width:33%;<b>Dispositivo</b></td>" +
"td style='width:33%;<b>Descripción</b></td><td style='width:33%; text-align: right;<b>Precio</b></td></tr>" + detalleDiagnostico +
"</table>" +
"table style='width:100%;<tr><td style='width:80%;</td><td style='width:20%;</td>" +
"table style='width:100%;<tr><td style='width: 60%;<b>Sub-Total</b></td>" +
"td style='text-align: right; width:40%;<b>suma</b></td></tr><tr><td><b>IVA</b></td>" +
"td style='text-align:right;<b>iivar</b></td></tr><tr><td><b>TOTAL</b></td>" +
"td style='text-align:right;color:red;<b>diagnostico.TotalDiagnostico</b></td></tr></table></td><td style='width:2%;</td></tr></table>" +
"</td></tr></table>" +
"<br><br><br><br><table style='width: 50%; class='clase_table'><tr><td style='width:2%;</td><td colspan='3'>" +
"table style='width: 100%; border='1'><tr style='background-color: gainsboro;'><td style='width: 8%;</td>" +
"td style='width:92%;<b>Una vez aceptado el diagnóstico se procederá a realizar la REPARACIÓN O MANTENIMIENTO de los dispositivos, conforme al precio establ</td></tr><tr><td style='width: 8%;<b>Rechazar</b></td>" +
"td style='width: 92%;<b>Una vez rechazado el diagnóstico se finalizará el proceso, informando al usuario el acercamiento para el retiro de los dispositivos</td></tr></table><br>" +
"</table></td></tr></table></tr>" +
"</tr><tr><td colspan='3'><br></td></tr></tr>" +
"td style='width: 20%;</td>" + fecha + "</td><td style='width: 60%; text-align: center;<b>Informe emitido por Sistema Web EPC-Compu</b></td>" +
"td style='width: 20%;</td></tr></table></table></tr>" +
"</tbody> </html> ";

var pdf = Pdf
.From(html)
.WithGlobalSetting("orientation", "portrait")
.WithObjectSetting("web.defaultEncoding", "utf-8")
.Content();
Session["download1"] = pdf;
Response.Redirect("IDescargarEstado.aspx", true);
}

```

**Figura 158.** Método de generación de informe de diagnóstico  
Elaborado por: Valeria Quispe

Como parte final se describe el método en caso de aceptación del diagnóstico.

```

private void ProcesoAceptacionDiagnostico(GridViewCommandEventArgs e)
{
    int estadoDiagnostico = 2;
    int index = Convert.ToInt32(e.CommandArgument);
    GridViewRow row = GridViewDiagnostico.Rows[index];
    int idDiagnostico = Convert.ToInt32(row.Cells[0].Text);
    EDiagnosticoCabecera diagnostico = new EDiagnosticoCabecera();
    diagnostico = NDiagnosticoCabecera.ActualizarEstadoDiagnostico(idDiagnostico, estadoDiagnostico);
    if (diagnostico == null)
    {
    }
    else
    {
        ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoID(diagnostico.IdSeleccionPertenece);
        EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroID(seleccion.IdIngresoRetiro);
        EAsignarRetiro asignacionRetiro = NAsignarRetiro.ObtenerAsignacionID(ingresoRetiro.IdAsignacionRetiro);

        EEstadoRetiro estado = new EEstadoRetiro();
        estado.IdEstadoRetiro = 0;
        estado.IdRetiroPertenece = asignacionRetiro.IdRetiroAsignacion;
        estado.FechaEstado = DateTime.Now.ToShortDateString();
        estado.EstadoRetiro = "Completado";
        estado.IdUsuarioResponsable = diagnostico.IdUsuarioDiagnostico;
        estado.NumEstadoRetiro = 6;
        estado = NEstadoRetiro.GuardarEstado(estado);

        LabelMensajeGrid.Text = "Registro Actualizado";
        EUsuario user = (EUsuario)Session["Usuario"];
        CargarListadoDiagnosticosXIdUsuario(user.IdUsuario);
        if (GridViewDiagnostico.Rows.Count == 0)
        {
            PanelAceptar.Visible = false;
            PanelRechazar.Visible = false;
        }
    }
}

```

**Figura 159.** Método de aceptación del diagnóstico en la capa de  
interfaz  
Elaborado por: Valeria Quispe

De la misma forma se detalla el método en caso de rechazo del diagnóstico

```

private void ProcesoRechazoDiagnostico(GridViewCommandEventArgs e)
{
    int estadoDiagnostico = 3;
    int index = Convert.ToInt32(e.CommandArgument);
    GridViewRow row = GridViewDiagnostico.Rows[index];
    int idDiagnostico = Convert.ToInt32(row.Cells[0].Text);
    EDiagnosticoCabecera diagnostico = new EDiagnosticoCabecera();
    diagnostico = NDiagnosticoCabecera.ActualizarEstadoDiagnostico(idDiagnostico, estadoDiagnostico);
    if (diagnostico == null)
    {
    }
    else
    {
        ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoID(diagnostico.IdSeleccionPertenece);
        EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroID(seleccion.IdIngresoRetiro);
        EAsignarRetiro asignacionRetiro = NAsignarRetiro.ObtenerAsignacionId(ingresoRetiro.IdAsignacionRetiro);

        EEstadoRetiro estado = new EEstadoRetiro();
        estado.IdEstadoRetiro = 0;
        estado.IdRetiroPertenece = asignacionRetiro.IdRetiroAsignacion;
        estado.FechaEstado = DateTime.Now.ToShortDateString();
        estado.EstadoRetiro = "Completado";
        estado.IdUsuarioResponsable = diagnostico.IdUsuarioDiagnostico;
        estado.NumEstadoRetiro = 6;
        estado = NEstadoRetiro.GuardarEstado(estado);

        LabelMensajeGrid.Text = "Registro Actualizado";
        EUsuario user = (EUsuario)Session["Usuario"];
        CargarListadoDiagnosticosXIdUsuario(user.IdUsuario);
        Response.Redirect("IAceptacionDiagnostico.aspx", true);
    }
}

```

**Figura 160.** Método de rechazo de diagnóstico en la capa de interfaz  
**Elaborado por:** Valeria Quispe

## Evidencia

### Subtarea Ver Diagnósticos

La subtarea de visualización de diagnósticos permite al usuario identificar los diagnósticos que tiene tanto si se encuentran pendientes o aceptados, todo a cabo de llevar un registro.



Visualización de Diagnósticos

Id Diagnóstico	Fecha	Total	Estado	Ver Diagnóstico
1	25/04/2022	13,44	Aceptado	
2	26/06/2022	24,64	Pendiente	

**Figura 161.** Visualización de Diagnósticos  
**Elaborado por:** Valeria Quispe

## Codificación

En la visualización de diagnósticos se emplea el mismo formato de la subtarea anterior para el informe y el llamado, por ello se presenta el método de listado de diagnósticos por usuario.



```

private void CargarListadoDiagnosticosxUsuario(int idUsuario)
{
    List<EDiagnosticoCabecera> diagnosticos = new List<EDiagnosticoCabecera>();
    diagnosticos = NDiagnosticoCabecera.ObtenerDiagnosticosTodosxIdUsuario(idUsuario);

    List<EVistaDiagnosticoTodos> vista = new List<EVistaDiagnosticoTodos>();
    foreach (var item in diagnosticos)
    {
        EVistaDiagnosticoTodos n = new EVistaDiagnosticoTodos();
        n.IdDiagnostico = item.IdDiagnostico;
        DateTime fecha = Convert.ToDateTime(item.FechaDiagnostico);
        string fecha1 = fecha.ToString("dd/MM/yyyy");
        n.FechaDiagnostico = fecha1;
        n.TotalDiagnostico = item.TotalDiagnostico;
        n.AceptacionDiagnostico = ObtenerEstadoAceptacion(item.AceptacionDiagnostico);
        vista.Add(n);
    }

    GridViewDiagnosticos.DataSource = vista;
    GridViewDiagnosticos.DataBind();
}

```

**Figura 162.** Método de recuperar diagnósticos por usuario en la capa de interfaz

**Elaborado por:** Valeria Quispe

## Tarea Completada



**Figura 163.** Gestión de Diagnósticos completada

**Elaborado por:** Valeria Quispe


**Prueba: HU08 Gestión de Diagnósticos**  
 en la lista [Pruebas](#)

Etiquetas  
 Prueba +

Descripción Editar

Llevar a cabo las pruebas con el fin de verificar el correcto funcionamiento de las operaciones en la gestión de diagnósticos

A Evaluar Ocultar los elementos marcados Eliminar

100% 

- Verificar la operación seleccionar ingreso, vista de ingresos, registro de selección
- Control de excepciones de seleccionar ingreso
- Verificar la operación ingresar diagnóstico, vista de selección de diagnósticos, agregado de información de diagnóstico, registro de diagnóstico
- Control de excepciones de ingresar diagnóstico
- Verificar la operación de aceptar diagnóstico, vista de diagnóstico completados, función de aceptar o rechazar, vista de informe de diagnóstico
- Control de excepciones de aceptar diagnóstico
- Verificar la operación ver diagnóstico, vista de listado de diagnóstico, vista de informe, diseño de informe
- Control de excepciones de ver diagnósticos

Añada un elemento

**Figura 164.** Pruebas de Gestión de Diagnósticos completada  
**Elaborado por:** Valeria Quispe

### H09 Menú de Diagnósticos

El menú de diagnóstico se enfocada en el desarrollo de la correcta navegación entre las funciones que componen la gestión de diagnósticos, estos habilitándose según el perfil de ingreso del usuario.



**Figura 165.** Menú de Diagnósticos  
**Elaborado por:** Valeria Quispe

### **Prueba**

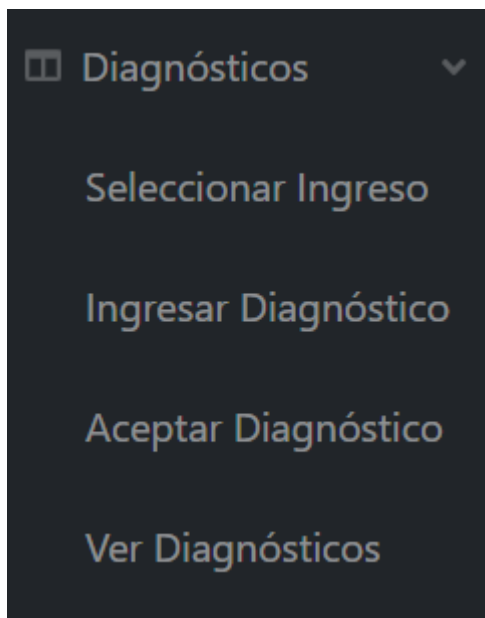
La finalidad de la prueba con respecto al menú de diagnóstico, es verificar la correcta habilitación de opciones de la gestión de diagnósticos conforme al perfil de ingreso del usuario, entre los cuales se puede definir como: secretaria, cliente, gerente, personal de entrega y recepción.



**Figura 166.** Pruebas del Menú de Diagnósticos  
**Elaborado por:** Valeria Quispe

### **Evidencia**

En las opciones del menú de diagnósticos se presenta las actividades que puede realizar los diferentes usuarios como lo son, seleccionar, ingresar, aceptar y ver diagnósticos.



**Figura 167.** Menú de Diagnósticos  
**Elaborado por:** Valeria Quispe

### Codificación

El desarrollo netamente se encuentra enfocado en la realización del menú en la capa de interfaz, donde se define las referencias a las diferentes páginas que componen la gestión de diagnósticos.

```
<a class="nav-link collapsed" href="#" data-bs-toggle="collapse" data-bs-target="#collapseLayoutsD" aria-expanded="false" aria
<div class="sb-nav-link-icon"><i class="fas fa-columns"></i></div>
Diagnósticos
<div class="sb-sidenav-collapse-arrow"><i class="fas fa-angle-down"></i></div>
/a>
:div class="collapse" id="collapseLayoutsD" aria-labelledby="headingOne" data-bs-parent="#sidenavAccordion">
<nav class="sb-sidenav-menu-nested nav">
<a class="nav-link" href="ISeleccionarIngreso.aspx">Seleccionar Ingreso</a>
<a class="nav-link" href="IIngresarDiagnostico.aspx">Ingresar Diagnóstico</a>
<a class="nav-link" href="IAceptacionDiagnostico.aspx">Aceptar Diagnóstico</a>
<a class="nav-link" href="IVerDiagnostico.aspx">Ver Diagnósticos</a>
</nav>
</div>
```

**Figura 168.** Redirección de formularios de gestión de diagnósticos  
**Elaborado por:** Valeria Quispe

Consecuentemente se presenta la activación del menú de diagnósticos, mismo que es llevado a cabo en la capa de interfaz.

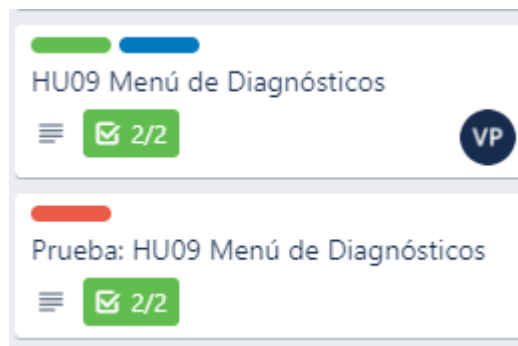
```

<nav class="sb-sidenav-menu-nested nav">
  <div id="opDiagnosticoTecnico">
    <a class="nav-link" href="ISeleccionarIngreso.aspx">Seleccionar Ingreso</a>
    <a class="nav-link" href="IIngresarDiagnostico.aspx">Ingresar Diagnóstico</a>
  </div>
  <div id="opDiagnosticoCliente">
    <a class="nav-link" href="IAceptacionDiagnostico.aspx">Aceptar Diagnóstico</a>
    <a class="nav-link" href="IVerDiagnostico.aspx">Ver Diagnósticos</a>
  </div>
</nav>

```

**Figura 169.** Habilitación de operación de gestión de diagnósticos por perfil  
**Elaborado por:** Valeria Quispe

### Tarea Completada



**Figura 170.** Menú de Diagnósticos completada  
**Elaborado por:** Valeria Quispe



**Figura 171.** Menú de Diagnósticos completada  
**Elaborado por:** Valeria Quispe

## **H10 Gestión de Mantenimiento y Reparación**

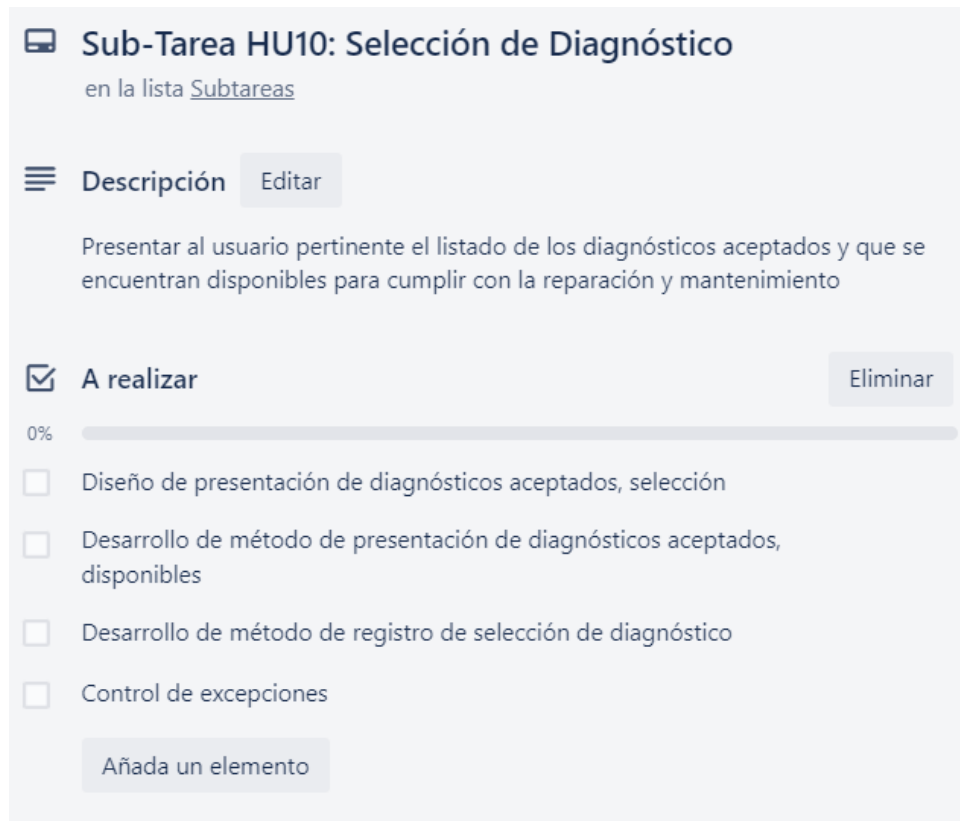
La gestión de reparación y mantenimiento se encuentra enfocado en la administración de los procesos que puede completar la reparación antes de ser finalizada, por ello con el fin de una mejor organización en la realización se divide en subtareas.

The screenshot shows a task management interface for 'HU10 Gestión de Reparación - Mantenimiento'. At the top, it indicates the task is in the 'Tareas por Hacer' list. Below this, there are sections for 'Miembros' (Members) and 'Etiquetas' (Tags). The 'Miembros' section shows a user 'VP' and a plus sign. The 'Etiquetas' section shows a blue tag labeled 'Pendiente' and another plus sign. A 'Descripción' (Description) section includes an 'Editar' (Edit) button and the text: 'Se estima el desarrollo de la gestión de reparación y mantenimiento con el fin de cumplir el servicio ofrecido por la empresa EPC-COMPU, donde se planifica la realización de ciertas operaciones'. Below the description is a checklist titled 'A realizar' (To be done) with a progress bar at 0% and an 'Eliminar' (Delete) button. The checklist items are: 'Desarrollar apartado de selección de diagnóstico', 'Desarrollar apartado de finalización de reparación/mantenimiento', 'Desarrollar apartado de visualización de estado de reparación/mantenimiento', and 'Desarrollar apartado de completar pago/entrega'. At the bottom of the checklist is a button labeled 'Añada un elemento' (Add an element).

**Figura 172.** Gestión de Reparación - Mantenimiento  
**Elaborado por:** Valeria Quispe

La tarea de selección de diagnóstico se enfoca en presentar un listado de diagnósticos aceptados y disponibles, con el fin de que el usuario pertinente pueda separarlo para realizar la previa reparación.

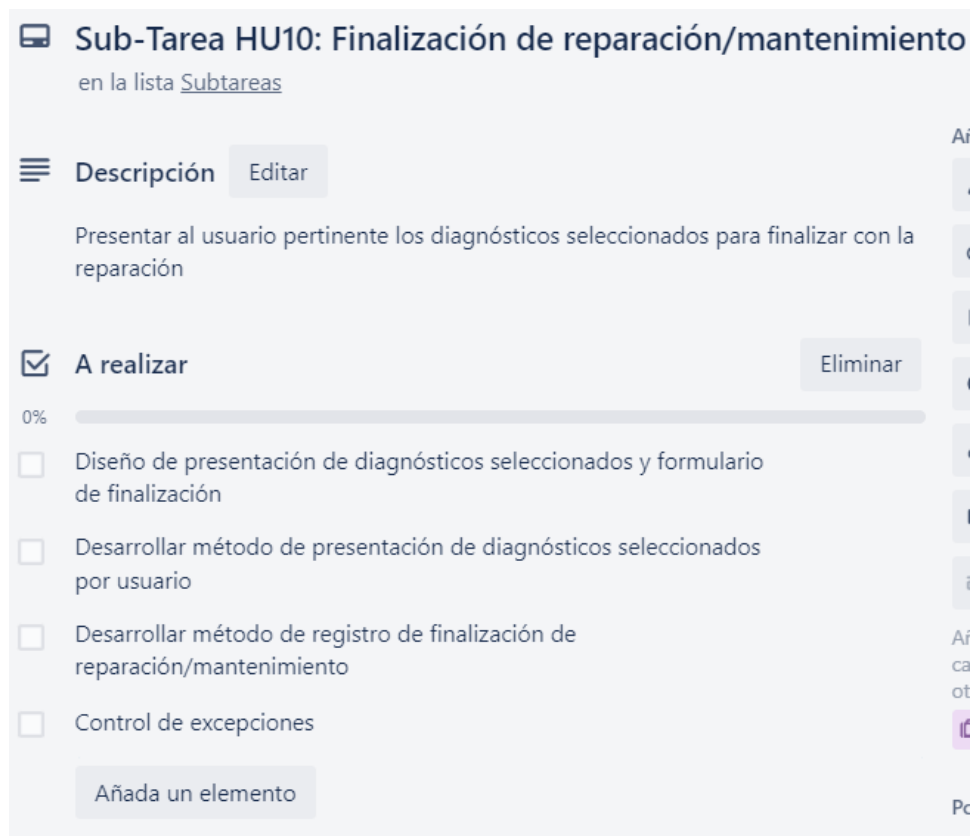




**Figura 173.** Sub-Tarea de la Historia de Usuario HU10 - Selección de Diagnóstico

**Elaborado por:** Valeria Quispe

La subtarea finalización de reparación y mantenimiento presenta al usuario técnico un listado de selecciones de diagnósticos, previamente seleccionado por el usuario de ingreso con ello podrá realizar la finalización de reparación o mantenimiento.



**Figura 174.** Sub-Tarea de la Historia de Usuario HU10 - Finalización de Reparación/Mantenimiento  
**Elaborado por:** Valeria Quispe

La subtarea ver estado de reparación o mantenimiento presenta al usuario pertinente el listado de las reparaciones en proceso, donde podrá acceder a la visualización del informe de estado, en el mismo podrá ver información de los estados que ha pasado con sus respectivas fechas.



**Figura 175.** Sub-Tarea de la Historia de Usuario HU10 - Ver estado de reparación/mantenimiento

**Elaborado por:** Valeria Quispe

La subtarea denominada completar pago y entrega cumple con la finalidad de visualizar las reparaciones o mantenimiento finalizados para que con el ello el personal encargado pueda completar el pago y la entrega de los dispositivos de interés.



**Figura 176.** Sub-Tarea de Historia de Usuario HU10 - Completar pago/entrega  
**Elaborado por:** Valeria Quispe

### Prueba

La finalidad de las pruebas es verificar el correcto funcionamiento de la gestión de mantenimiento y reparación, en ello evaluará la funcionalidad de selección, finalización, visualización de estado y completar entrega.



**Figura 177.** Pruebas de Gestión de Reparación/Mantenimiento  
**Elaborado por:** Valeria Quispe

### Subtarea Selección de diagnóstico

#### Evidencia

Se presenta la interfaz de selección de diagnóstico, donde el usuario podrá observar datos de la aceptación como la fecha y el valor por el cual se planteó de la reparación.

**Selección de Diagnóstico**

LISTADO DE DIAGNÓSTICOS ☰

Id Diagnóstico	Fecha	Total	Ver Diagnóstico
	2 26/06/2022	24,64	▶

**Figura 178.** Vista de diagnóstico aceptados y disponibles  
**Elaborado por:** Valeria Quispe

Consecuente a ello se visualizará el formulario de registro de selección de diagnóstico, en el mismo especificará un estimado de tiempo para la reparación observando conjuntamente la información del cliente y el diagnóstico realizado por el personal técnico.

**Selección de Diagnóstico**  
Visualización de Retiro

---

INFORMACIÓN DEL RETIRO: 📄

Id Retiro:	1	Nombre del Cliente:	Valeria Quispe
Dirección:	Ciudadela cumanda Alrededor	Teléfono:	0979398197
Día - Hora Solicitado:	Jueves-15:00	Referencia Domicilio:	Frente de upc
Total Diagnóstico:	24,64	Tiempo Estimado de Reparación:	0

---

DETALLE DE DISPOSITIVOS TECNOLÓGICOS 📄

Dispositivo	Descripción	Diagnóstico
CPU-DELL	Ninguna	Necesita actualización de software
Impresora-EPSON	Manuales	Necesita actualización de software

---

Seleccionar
Regresar

**Figura 179.** Formulario de registro de selección de diagnóstico  
**Elaborado por:** Valeria Quispe

### Codificación

En la presentación del listado de diagnósticos aceptados y que se encuentra disponibles se desarrolla el método cargarListadoDiagnosticoAceptadosLibres(), enfocado en recuperar los diagnósticos a través del método de la capa de negocio.

```
private void cargarListadoDiagnosticosAceptadosLibres()
{
    List<EDiagnosticoCabecera> listadiagnostico = new List<EDiagnosticoCabecera>();
    int estadoDiagnostico = 2;
    int estadoSeleccion = 0;
    listadiagnostico = NDiagnosticoCabecera.ObtenerDiagnosticosAceptadosLibres(estadoSeleccion, estadoDiagnostico);
    List<EDiagnosticoCabecera> listaLimpia = new List<EDiagnosticoCabecera>();
    foreach (var item in listadiagnostico)
    {
        EDiagnosticoCabecera diagnostico = new EDiagnosticoCabecera();
        diagnostico.IdDiagnostico = item.IdDiagnostico;
        DateTime fecha = Convert.ToDateTime(item.FechaDiagnostico);
        string fecha1 = fecha.ToString("dd/MM/yyyy");
        diagnostico.FechaDiagnostico = fecha1;
        diagnostico.TotalDiagnostico = item.TotalDiagnostico;
        listaLimpia.Add(diagnostico);
    }

    GridViewDiagnostico.DataSource = listaLimpia;
    GridViewDiagnostico.DataBind();
}
```

**Figura 180.** Método de obtener diagnósticos aceptados libres en la  
capa de interfaz  
**Elaborado por:** Valeria Quispe

Y una vez que el usuario de interés se encuentre interesado en un registro, él lo seleccionará donde se procede con la recuperación de la información del diagnóstico, así como la de los dispositivos, facilitando al personal a estimar un tiempo para la reparación.

```
private void cargarDiagnostico(int idDiagnostico)
{
    EDiagnosticoCabecera diagnostico = NDiagnosticoCabecera.ObtenerDiagnosticoxID(idDiagnostico);
    ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoID(diagnostico.IdSeleccionPertenece);
    EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroxID(seleccion.IdIngresoRetiro);
    EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxID(ingresoRetiro.IdAsignacionRetiro);
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);

    List<EDetalleRetiro> listadoArticulos = NCabeceraRetiro.ObtenerDetalleRetiroxID(retiro.IdRetiro);
    EUsuario usuario = MUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
    EPersona cliente = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);

    TextBoxIdRetiro.Text = retiro.IdRetiro.ToString();
    TextBoxDireccion.Text = cliente.DireccionPersona;
    TextBoxDiaHora.Text = retiro.DiaRetiro + "-" + retiro.HoraRetiro;
    TextBoxNombreCliente.Text = cliente.NombrePersona + " " + cliente.ApellidoPersona;
    TextBoxTelefono.Text = cliente.TelefonoPersona;
    TextBoxReferenciaDomi.Text = retiro.ReferenciaDomicilio;
    LabelIdRetiro.Text = (idDiagnostico).ToString();
    TextBoxTotal.Text = diagnostico.TotalDiagnostico.ToString();
    List<EDiagnosticoDetalle> listadDetalleDiagnostico = NDiagnosticoCabecera.ObtenerDetalleDiagnosticoxID(diagnostico.IdDiagnostico);
    List<EVistaDiagnostico> listaVista = new List<EVistaDiagnostico>();
    foreach (var item in listadDetalleDiagnostico)
    {
        EDiagnosticoDetalle detalle = NDiagnosticoCabecera.ObtenerDetalleDiagnosticoSoloIdDetalle(item.IdDiagnosticoDetalle);
        EDetalleRetiro detalleRe = NCabeceraRetiro.ObtenerDetalleSoloIdDetalle(detalle.IdRetiroDetallePertenece);
        EVistaDiagnostico vista = new EVistaDiagnostico();
        vista.Dispositivo = detalleRe.NombreArticulo + "-" + detalleRe.MarcaArticulo;
        vista.Descripcion = detalleRe.DescripcionArticulo;
        vista.Diagnostico = detalle.DescripcionDiagnostico;
        listaVista.Add(vista);
    }
    GridViewDispositivos.DataSource = listaVista;
    GridViewDispositivos.DataBind();
}
}
```

**Figura 181.** Método de recuperar datos de diagnóstico en la capa de interfaz

**Elaborado por:** Valeria Quispe

Consecuente a finalizar la selección del diagnóstico se desarrolla el método de actualizar estado de diagnóstico, donde genera el registro de reparación y actualiza los diferentes estados.

```
private void ActualizarEstadoDiagnostico()
{
    //Cambiar Estado Diagnostico
    int idDiagnostico = Convert.ToInt32(LabelIdRetiro.Text);
    int estadoDiagnostico = 4;
    int estadoSeleccion = 1;
    EDiagnosticoCabecera diagnostico = NDiagnosticoCabecera.ActualizarEstadoDiagnosticoSeleccion(idDiagnostico, estadoDiagnostico, estadoSeleccion);

    EUsuario user = (EUsuario)Session["Usuario"];
    //Guardar Reparación
    EReparacion reparacion = new EReparacion();
    reparacion.IdReparacion = 0;
    reparacion.IdUsuarioRepa = user.IdUsuario;
    reparacion.FechaReparacion = DateTime.Now.ToShortDateString();
    reparacion.TiempoEstimado = Convert.ToInt32(TextBoxDiasEstimados.Text);
    reparacion.IdDiagnosticoPertenece = idDiagnostico;
    reparacion.EstadoReparacion = 1;
    reparacion.EstadoEliminacion = 0;
    reparacion.Observacion = "Ninguna";
    reparacion = NReparacion.GuardarReparacion(reparacion);

    if (reparacion == null)
    {
    }
    else
    {
        //Crear Estado Iniciado
        EEstadoReparacion iniciado = new EEstadoReparacion();
        iniciado.IdEstadoReparacion = 0;
        iniciado.IdUsuarioResponsable = user.IdUsuario;
        iniciado.FechaEstado = DateTime.Now.ToShortDateString();
        iniciado.IdReparacionPertenece = reparacion.IdReparacion;
        iniciado.EstadoReparacion = "Iniciado";
        iniciado.NumEstado = 1;
        iniciado = NReparacion.GuardarEstadoReparacion(iniciado);
    }
}
```

```

//Crear Estado Seleccionado
EEstadoReparacion seleccionado = new EEstadoReparacion();
seleccionado.IdEstadoReparacion = 0;
seleccionado.IdUsuarioResponsable = user.IdUsuario;
seleccionado.FechaEstado = DateTime.Now.ToShortDateString();
seleccionado.IdReparacionPertenece = reparacion.IdReparacion;
seleccionado.EstadoReparacion = "Seleccionado";
seleccionado.NumEstado = 2;
seleccionado = NReparacion.GuardarEstadoReparacion(seleccionado);

//Crear En Proceso
EEstadoReparacion proceso = new EEstadoReparacion();
proceso.IdEstadoReparacion = 0;
proceso.IdUsuarioResponsable = user.IdUsuario;
proceso.FechaEstado = DateTime.Now.ToShortDateString();
proceso.IdReparacionPertenece = reparacion.IdReparacion;
proceso.EstadoReparacion = "En Proceso";
proceso.NumEstado = 3;
proceso = NReparacion.GuardarEstadoReparacion(proceso);

//Actualizar Estado Reparación
EReparacion repActualizar = new EReparacion();
int estado = 3;
repActualizar = NReparacion.ActualizarEstadoReparacion(reparacion.IdReparacion, estado);

LabelMensajeRegistro.Text = "Registro Guardado Correctamente";
cargarListadoDiagnosticosAceptadosLibres();
MultiView1.ActiveViewIndex = 1;
}
}

```

**Figura 182.** Método de guardar selección de diagnóstico en la capa de interfaz  
**Elaborado por:** Valeria Quispe

### Subtarea Finalizar Reparación

#### Evidencia

Para finalizar con la reparación se indica al usuario el listado de las selecciones de diagnósticos a su cargo, en ello podrá visualizar la fecha de selección y el tiempo estimado de reparación.

**Reparaciones a Cargo**

LISTADO DE DIAGNÓSTICOS ☰

Id Reparación	Fecha	Tiempo Estimado(Días)	Ver Reparación
	2 04/07/2022	5	▶

**Figura 183.** Vista de diagnósticos seleccionados por usuario  
**Elaborado por:** Valeria Quispe

Conforme a cumplir con la finalización se presenta el formulario, donde el personal técnico tendrá la posibilidad de agregar cualquier observación en la reparación o si es el caso ninguno indicará que se realizó sin ningún inconveniente.



## Reparaciones a Cargo Visualización de Reparación-Mantenimiento

INFORMACIÓN DEL RETIRO:

Id Retiro:	1	Nombre del Cliente:	Valeria Quispe
Dirección:	Ciudadela cumanda Alrededor	Teléfono:	0979398197
Día - Hora Solicitado:	Jueves-15:00	Referencia Domicilio:	Frente de upc
Total Diagnóstico:	24,64	Observación:	

DETALLE DE DISPOSITIVOS TECNOLÓGICOS

Dispositivo	Descripción	Diagnóstico
CPU-DELL	Ninguna	Necesita actualización de software
Impresora-EPSON	Manuales	Necesita actualización de software

[Finalizar](#) [Regresar](#)

**Figura 184.** Formulario de finalización de reparación  
**Elaborado por:** Valeria Quispe

### Codificación

Conforme a cumplir con la finalización de reparación, se desarrolla el método encargado de recuperar un listado de diagnósticos seleccionados por un usuario de perfil técnico esto para que pueda visualizar información previa y de interés.

```
private void cargarListadoReparacionesSeleccionados(int idUsuario)
{
    List<EReparacion> listaReparacion = new List<EReparacion>();
    int estadoRe = 3;
    listaReparacion = NReparacion.ObtenerRepaxIdUsuarioEstado(idUsuario,estadoRe);

    List<EReparacion> listaVista = new List<EReparacion>();
    foreach (var item in listaReparacion)
    {
        EReparacion rep = new EReparacion();
        rep.IdReparacion = item.IdReparacion;
        rep.TiempoEstimado = item.TiempoEstimado;
        DateTime fecha = Convert.ToDateTime(item.FechaReparacion);
        string fecha1 = fecha.ToString("dd/MM/yyyy");
        rep.FechaReparacion = fecha1;
        listaVista.Add(rep);
    }
    GridViewReparacion.DataSource = listaVista;
    GridViewReparacion.DataBind();
}
```

**Figura 185.** Método de obtener reparaciones iniciado por usuario en  
la capa de interfaz  
**Elaborado por:** Valeria Quispe

Una vez que el personal técnico este por terminar la reparación se procede a seleccionarlo, con ello se recupera la información de interés a través de método cargarReparacion().

```
private void cargarReparacion(int idReparacion)
{
    EReparacion reparacion = NReparacion.ObtenerReparacionxID(idReparacion);
    EDiagnosticoCabecera diagnostico = NDiagnosticoCabecera.ObtenerDiagnosticoxID(reparacion.IdDiagnosticoPertenece);
    ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoxID(diagnostico.IdSeleccionPertenece);
    EIngresoRetiro ingresoRetiro = NIngresoRetiro.ObtenerIngresoRetiroxID(seleccion.IdIngresoRetiro);
    EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxID(ingresoRetiro.IdAsignacionRetiro);
    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);
    List<EDetalleRetiro> listadoArticulos = NCabeceraRetiro.ObtenerDetalleRetiroxID(retiro.IdRetiro);
    EUsuario usuario = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
    EPersona cliente = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);
    TextBoxIdRetiro.Text = retiro.IdRetiro.ToString();
    TextBoxDireccion.Text = cliente.DireccionPersona;
    TextBoxDiaHora.Text = retiro.DiaRetiro + "-" + retiro.HoraRetiro;
    TextBoxNombreCliente.Text = cliente.NombrePersona + " " + cliente.ApellidoPersona;
    TextBoxTelefono.Text = cliente.TelefonoPersona;
    TextBoxReferenciaDomi.Text = retiro.ReferenciaDomicilio;
    LabelIdRetiro.Text = (idReparacion).ToString();
    TextBoxTotal.Text = diagnostico.TotalDiagnostico.ToString();
    List<EDiagnosticoDetalle> listadDetalleDiagnostico = NDiagnosticoCabecera.ObtenerDetalleDiagnosticoxID(diagnostico.IdDiagnostico);
    List<EVistaDiagnostico> listaVista = new List<EVistaDiagnostico>();
    foreach (var item in listadDetalleDiagnostico)
    {
        EDiagnosticoDetalle detalle = NDiagnosticoCabecera.ObtenerDetalleDiagnosticoSoloxIdDetalle(item.IdDiagnosticoDetalle);
        EDetalleRetiro detalleRe = NCabeceraRetiro.ObtenerDetalleSoloxID(detalle.IdRetiroDetallePertenece);
        EVistaDiagnostico vista = new EVistaDiagnostico();
        vista.Dispositivo = detalleRe.NombreArticulo + "-" + detalleRe.MarcaArticulo;
        vista.Descripcion = detalleRe.DescripcionArticulo;
        vista.Diagnostico = detalle.DescripcionDiagnostico;
        listaVista.Add(vista);
    }
    GridViewDispositivos.DataSource = listaVista;
    GridViewDispositivos.DataBind();
}
```

**Figura 186.** Método de cargar reparación en la capa de interfaz  
**Elaborado por:** Valeria Quispe

Para finalizar la reparación se lleva a cabo el desarrollo del método MetFinalizarReparacion(), encargado del registro y la actualización de los nuevos estados.

```

private void MetFinalizarReparacion()
{
    EUsuario us = new EUsuario();
    us = (EUsuario)Session["Usuario"];
    //cambiar Estado en Reparacion
    EReparacion repActualizar = new EReparacion();
    int estado = 4;
    repActualizar = NReparacion.ActualizarEstadoReparacion(Convert.ToInt32(LabelIdRetiro.Text), estado);
    //cambiar observacion
    string obs = "";
    if (TextBoxObservacion.Text == "")
    {
        obs = "Ninguna";
    }else{
        obs = TextBoxObservacion.Text;
    }
    EReparacion repActualizarOBS = new EReparacion();
    repActualizarOBS = NReparacion.ActualizarObsReparacion(Convert.ToInt32(LabelIdRetiro.Text), obs);
    //Guardar Nuevo Estado-Finalizado
    EEstadoReparacion proceso = new EEstadoReparacion();
    proceso.IdEstadoReparacion = 0;
    proceso.IdUsuarioResponsable = us.IdUsuario;
    proceso.FechaEstado = DateTime.Now.ToShortDateString();
    proceso.IdReparacionPertenece = Convert.ToInt32(LabelIdRetiro.Text);
    proceso.EstadoReparacion = "Finalizado";
    proceso.NumEstado = 4;
    proceso = NReparacion.GuardarEstadoReparacion(proceso);
    //Generar Finalizar
    EReparacionCompleta completo = new EReparacionCompleta();
    completo.IdReparacionCompleta = 0;
    completo.FechaReparacionCompleta = DateTime.Now.ToShortDateString();
    completo.EntregaDomicilio = 0;
    completo.PagoCompletado = 0;
    completo.EstadoReparacionCompleto = 1;
    completo.IdReparacionPertenece = Convert.ToInt32(LabelIdRetiro.Text);
    completo = NReparacion.GuardarReparacionCompleto(completo);
    //guardar en la nueva tabla
    cargarListadoReparacionesSeleccionados(us.IdUsuario);
    MultiView1.ActiveViewIndex = 1;
    LabelMensajeRegistro.Text = "Se finalizo de manera correcta el registro";
}

```

**Figura 187.** Método de finalizar reparación en la capa de interfaz  
**Elaborado por:** Valeria Quispe

## Subtarea Estado de Reparaciones

### Evidencia

Para la visualización del estado de la reparaciones o mantenimiento en proceso se presenta un listado donde se detalla la fecha, tiempo estimado de reparación, valor y el estado actual en cual se encuentra.



Estado de Reparaciones

Id Reparación	Fecha Solicitada	Tiempo Estimado	Total	Estado Actual	Ver Detalle
1	19/05/2022	3	13,44	Finalizado	

**Figura 188.** Vista de reparaciones en proceso  
**Elaborado por:** Valeria Quispe

Y para que el usuario de interés pueda visualizar seleccionará ver detalle donde se genera el informe con toda la información pertinente a la reparación y al historial.



## EPC-COMPU

RICARDO PÉREZ GERENTE

Dirección: La Quigo-Atocha

Mantenimiento y Reparación de Equipos Electrónicos



### ESTADOS DE LA REPARACIÓN N°1

#### INFORMACIÓN DEL CLIENTE

<b>Nombre:</b>	Valeria Quispe	<b>RUC / CI:</b>	1805153580
<b>Dirección:</b>	Ciudadela curamda Alrededor	<b>Fecha Solicitada:</b>	29/03/2022
<b>Teléfono:</b>	0979398197	<b>Día y Hora estimada:</b>	Jueves-17:00
<b>Referencia:</b>	Frete de upc	<b>Retiro:</b>	3

#### ARTÍCULOS DE REPARACIÓN

Nombre	Marca	Modelo	Descripción	Servicio
CPU	DELL	C.00	Manuales	Mantenimiento

#### DIAGNÓSTICO DE DEPOSITIVOS

Dispositivo	Descripción	Precio
CPU-DELL	Necesita actualización de software	12

Sub-Total: 12

IVA: 1,44

**TOTAL: 13,44**

#### ESTADOS DE LA REPARACIÓN

Id Estado	Fecha	Estado
1	19/05/2022	Iniciado
2	19/05/2022	Seleccionado
3	19/05/2022	En Proceso
5	24/05/2022	Finalizado

04/07/2022

Informe emitido por Sistema Web EPC-Compu

**Figura 189.** Informe de estado de reparación  
**Elaborado por:** Valeria Quispe

## Codificación

La presentación de reparaciones se lo realiza a través del desarrollo del método `CargarListadoReparaciones(idUsuario)`, encargado de recuperar el listado de reparaciones pertenecientes a un usuario.

```

private void CargarListadoReparaciones(int idUsuario)
{
    List<EVistaEstadoReparacion> listaVista = new List<EVistaEstadoReparacion>();
    listaVista = NReparacion.ObtenerListaReparacionesxIdUsuario(idUsuario);

    List<EVistaEstadoReparacion> listaVistaN = new List<EVistaEstadoReparacion>();
    foreach (var item in listaVista)
    {
        EVistaEstadoReparacion n = new EVistaEstadoReparacion();
        n.IdReparacion = item.IdReparacion;
        DateTime fecha = Convert.ToDateTime(item.FechaReparacion);
        string fecha1 = fecha.ToString("dd/MM/yyyy");
        n.FechaReparacion = fecha1;
        n.TiempoEstimado = item.TiempoEstimado;
        n.TotalDiagnostico = item.TotalDiagnostico;
        n.Estado = RetornarEstado(item.EstadoReparacion);
        listaVistaN.Add(n);
    }

    GridViewReparaciones.DataSource = listaVistaN;
    GridViewReparaciones.DataBind();
}

```

**Figura 190.** Método de obtención de reparaciones en proceso por usuario en la capa de interfaz  
**Elaborado por:** Valeria Quispe

En la presentación del informe el método de generación es similar al método de estado de retiros y con el fin de preservar la información necesaria se destaca el apartado con mayor importancia.

```

private void generarPDFReparaciones(int idReparacion)
{
    EReparacion reparacion = new EReparacion();
    reparacion = NReparacion.ObtenerReparacionxID(idReparacion);
    EDiagnosticoCabecera diagnostico = new EDiagnosticoCabecera();
    diagnostico = NDiagnosticoCabecera.ObtenerDiagnosticoxID(reparacion.IdDiagnosticoPertenece);
    ESeleccionIngreso seleccion = NSeleccionIngreso.ObtenerSeleccionIngresoxID(diagnostico.IdSeleccionPertenece);
    EIngresoRetiro ingreso = NIngresoRetiro.ObtenerIngresoRetiroxID(seleccion.IdIngresoRetiro);
    EAsignarRetiro asignacion = NAsignarRetiro.ObtenerAsignacionxId(ingreso.IdAsignacionRetiro);

    ECabeceraRetiro retiro = NCabeceraRetiro.ObtenerRetiroxID(asignacion.IdRetiroAsignacion);
    DateTime fechaSolicitada = Convert.ToDateTime(retiro.FechaSolicitada);
    string fechaSol = fechaSolicitada.ToString("dd/MM/yyyy");
    EUsuario usuario = NUsuario.DevolverUsuarioxId(retiro.IdUsuarioRetiro);
    EPersona persona = NPersona.DevolverPersonaxId(usuario.IdPersonaPertenece);
    List<EDetalleRetiro> listaDetalleRetiro = NCabeceraRetiro.ObtenerDetalleRetiroxID(retiro.IdRetiro);
    string detalleRetiro = "";
    foreach (var item in listaDetalleRetiro)
    {
        detalleRetiro += "<tr><td style='width:18%;>" + item.NombreArticulo + "</td><td style='width:18%;>" + item.MarcaArticulo + "</td>" +
            "<td style='width:18%;>" + item.ModeloArticulo + "</td><td style='width:24%;>" + item.DescripcionArticulo + "</td>" +
            "<td style='width:18%;>" + item.TipoServicio + "</td></tr>";
    }
}

```

**Figura 191.** Apartado de obtención de información de reparación en la capa de interfaz  
**Elaborado por:** Valeria Quispe

**Tarea Completada**



**Figura 192.** Gestión de Reparación-Mantenimiento completada  
**Elaborado por:** Valeria Quispe



**Figura 193.** Gestión de Reparación/Mantenimiento terminada  
**Elaborado por:** Valeria Quispe

### **H11 Menú de Mantenimiento y Reparación**

El menú de mantenimiento y reparación se enfocada en el desarrollo de la correcta navegación entre las funciones que componen la gestión de reparación, estos habilitándose según el perfil de ingreso del usuario.



**Figura 194.** Menú de Reparación-Mantenimiento  
**Elaborado por:** Valeria Quispe

### **Prueba**

La finalidad de la prueba con respecto al menú de reparación y mantenimiento, es verificar la correcta habilitación de opciones de la gestión de reparaciones conforme al perfil de ingreso del usuario, entre los cuales se puede definir como: secretaria, cliente, gerente, personal de entrega y recepción.

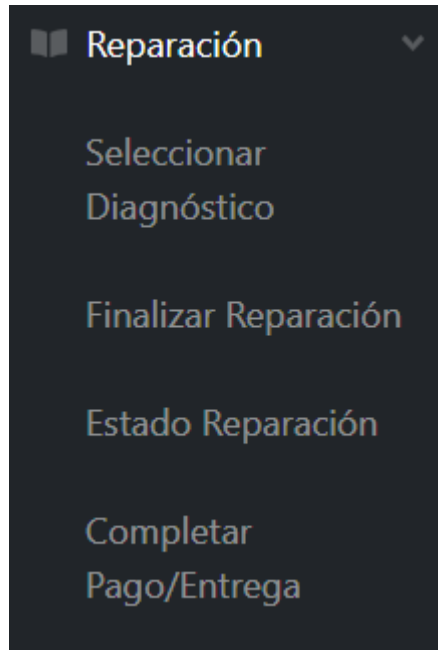




**Figura 195.** Pruebas para el Menú de Reparación-Mantenimiento  
**Elaborado por:** Valeria Quispe

### **Evidencia**

En las opciones del menú de reparaciones se presenta las actividades que puede realizar los diferentes usuarios como lo son, seleccionar diagnóstico, finalizar reparación, visualizar estado de reparación y completar pago.



**Figura 196.** Menú de Reparación/Mantenimiento en el sistema  
**Elaborado por:** Valeria Quispe

### Codificación

El desarrollo netamente se encuentra enfocado en la realización del menú en la capa de interfaz, donde se define las referencias a las diferentes páginas que componen la gestión de reparaciones.

```
<div class="collapse" id="collapseLayoutsRM" aria-labelledby="headingOne" data-bs-parent="#sidenavAccordion">
  <nav class="sb-sidenav-menu-nested nav">
    <a class="nav-link" href="/ISeleccionarDiagnostico.aspx">Seleccionar Diagnóstico</a>
    <a class="nav-link" href="/IFinalizarReparacion.aspx">Finalizar Reparación</a>
    <a class="nav-link" href="/IEstadoReparacion.aspx">Estado Reparación</a>
    <a class="nav-link" href="/IEstadoReparacion.aspx">Completar Pago/Entrega</a>
  </nav>
</div>
```

**Figura 197.** Redirección de formularios de gestión de reparación  
**Elaborado por:** Valeria Quispe

Consecuentemente se presenta la activación del menú de reparaciones, mismo que es llevado a cabo en la capa de interfaz.

```

<nav class="sb-sidenav-menu-nested nav">
  <div id="opReparacionTecnico">
    <a class="nav-link" href="ISeleccionarDiagnostico.aspx">Seleccionar Diagnóstico</a>
    <a class="nav-link" href="IFinalizarReparacion.aspx">Finalizar Reparación</a>
  </div>
  <div id="opReparacionCliente">
    <a class="nav-link" href="IEstadoReparacion.aspx">Estado Reparación</a>
  </div>
  <div id="opReparacionSecretaria">
    <a class="nav-link" href="ICompletarReparacion.aspx">Completar Pago/Entrega</a>
  </div>
</nav>

```

División de operaciones de gestión de reparación por perfil

### Tarea Completada



**Figura 198.** Menú de Reparación - Mantenimiento completada  
**Elaborado por:** Valeria Quispe



**Figura 199.** Menú de Reparación - Mantenimiento completada  
**Elaborado por:** Valeria Quispe

## H12 Gestión de Reportes

La gestión de reportes se enfoca en la presentación de informes que ayuden a los usuarios a tomar decisiones, en ello presentar la utilidad de la utilización del Sistema Web EPC-COMPU



**Figura 200.** Gestión de Reportes  
**Elaborado por:** Valeria Quispe

## Prueba

El fin de la prueba es evaluar el correcto funcionamiento en la generación de los informes establecidos, a su vez ayuda a ver si la información expuesta es la adecuada para el usuario.



**Figura 201.** Pruebas para la Gestión de Reportes  
**Elaborado por:** Valeria Quispe

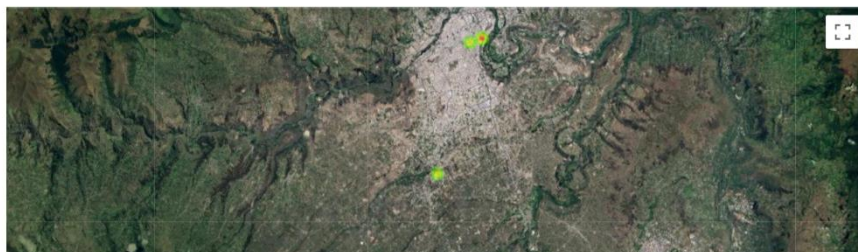
### **Evidencia**

Un apartado a la presentación de informes, se destaca la visualización del mapa de calor de la ubicación de los clientes, con ello facilitar a propietario en la toma de decisiones en si ampliar su negocio en ciertos puntos estratégicos.

## Visualización de Mapa de Calor

INFORMACIÓN ADICIONAL

Total de Clientes : 4



**Figura 202.** Mapa de calor de clientes  
**Elaborado por:** Valeria Quispe


De la misma forma se presenta en diseño de página que se encontrara a cargo de gestionar los informes pertenecientes a cada perfil de interés.

# Gestión de Informes




**Figura 203.** Diseño de presentación de opciones de informes  
**Elaborado por:** Valeria Quispe

Se presenta el diseño de algunos informes, para el aprecio y visualización



**EPC-COMPU**  
RICARDO PÉREZ GERENTE  
Dirección: La Quigo-Atocha  
Mantenimiento y Reparación de Equipos Electrónicos



---

**DETALLE DE TOTAL DE REPARACIONES COMPLETADAS**

**Nota:** El informe presenta el ID de las reparaciones completadas, con previa descripción del pago (Pendiente / Completado -> en caso de estar cancelado o acordar algún medio de pago con el cliente) y así se detalle si la entrega se encuentra pendiente o se realizará en su domicilio

**DETALLE DE REPARACIONES**

Id	Fecha Completada	Pago	Entrega	Total
1	24/05/2022	Pendiente	Pendiente	13,44

---

29/06/2022 Informe emitido por Sistema Web EPC-Compu

**Figura 204.** Informe de reparaciones totales  
**Elaborado por:** Valeria Quispe



## Codificación

En la visualización del mapa de calor se destaca la recuperación de las coordenadas geográficas, en este caso la ubicación de los domicilios de los clientes como se detalla en el siguiente método.

```
public string RecuperarValores()
{
    DataTable dt = new DataTable();
    string datos;

    datos = "[";
    using (SqlConnection con = new SqlConnection(sql))
    {
        using (SqlCommand cmd = new SqlCommand("select latitud,longitud from RetiroCabecera", con))
        {
            con.Open();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            da.Fill(dt);
            foreach (DataRow col in dt.Rows)
            {
                datos = datos + "new google.maps.LatLng(";
                datos = datos + col[0] + "," + col[1];
                datos = datos + "),";
            }
            datos = datos + "];";
        }
    }
    return datos;
}
```

**Figura 205.** Método de recuperar coordenadas geográficas de clientes en la capa de datos

**Elaborado por:** Valeria Quispe

Consecuente a ello se realiza el tratamiento de los datos recuperados por medio de JavaScript, a su vez es plasmado con la librería de Google Maps para que el usuario pueda tener la visualización del mapa de calor.

```

<script type="text/javascript">

function initialize() {
    var posiciones = <%=RecuperarValores()%>;
    var primeraposicion = <%=RecuperarValoresPrimerPosicion() %>;
    // Configuración del mapa

    // Agregando el mapa al tag de id googleMap

    var mapProp = {
        zoom: 12,
        center: primeraposicion,
        mapTypeId: 'satellite'
    };
    // Agregando el mapa al tag de id googleMap
    var map = new google.maps.Map(document.getElementById("modal"), mapProp);
    var heatmapData = posiciones;

    var heatmap = new google.maps.visualization.HeatmapLayer({
        data: heatmapData
    });
    heatmap.setMap(map);
}
// Inicializando el mapa cuando se carga la página
google.maps.event.addDomListener(window, 'load', initialize);

</script>

```

**Figura 206.** Método de visualización de mapa de calor en la capa de interfaz

**Elaborado por:** Valeria Quispe

En la presentación de los informes respectivos a los usuarios, el diseño es llevado a cabo en HTML y codificación en C#, donde se presenta un pequeño apartado del diseño ya que los restantes informes cumplen un mismo desarrollo.

```

private void GenerarInformeDetalledeReparacionesCompletadas()
{
    List<VistaReparacionesCompleta> listaReparaciones = IRreparacion.ObtenerListaReparacionesCompletas();
    string detalleRetiro = "";
    foreach (var item in listaReparaciones)
    {
        DateTime fechaF = Convert.ToDateTime(item.FechaReparacion);
        string fechaSol = fechaF.ToString("dd/MM/yyyy");
        detalleRetiro += "<tr><td style='width:18%;>" + item.IdReparacion + "</td><td style='width:18%;>" + fechaSol + "</td>" +
            "<td style='width:18%;>" + devolverPago(item.Pago) + "</td><td style='width:24%;>" + devolverEntrega(item.EntregaDomicilio) + "</td>" +
            "<td style='width:18%;>" + item.Total + "</td></tr>";
    }
    string fecha = DateTime.Now.ToShortDateString();
    string html = "<!DOCTYPE html>" +
        "<html><head><meta charset='UTF-8'><title> Informe Número de Reparaciones Completadas </title>" +
        "</head><style>.clase_table {border-collapse: separate;border-spacing: 10px;border: 1px solid black;border-radius: 15px;-moz-border-radius: 20px;padding: 2px;}" +
        "</style></head><body><table border='1' style='width:100%;>" +
        "<tbody><tr><td colspan='5' style='text-align:center;><img src='https://i.ibb.co/S54jnGS/logo.png' width='100px' height='100px'></td>" +
        "<td style='width:80%; text-align:center;><h1 style='text-align:center;>EPC-COMPU </h1><p>RICARDO PÉREZ GERENTE</p><p>Dirección: La Quigo-Atocha </p></td>" +
        "<td style='width:10%; text-align:center;><img src='https://i.ibb.co/mq01trnd/Logo-Ambato.jpg' width='100px' height='100px'></td></tr>" +
        "<tr><td colspan='5' style='text-align:center;><h3>DETALLE DE TOTAL DE REPARACIONES COMPLETADAS </h3></td>" +
        "<td colspan='5'></td></tr>" +
        "</tbody></table>" +
        "<table border='1' style='width:100%;>" +
        "<tbody><tr><td colspan='5' style='text-align:center;><b>Nota:</b>" +
        "<td colspan='5' style='text-align:center;><b>El informe presenta el ID de las reparaciones completadas, con previa descripción del pago (Pendiente /Completado -> en caso de estar cancelado o acordar al" +
        "<td colspan='5' style='text-align:center;></td></tr></tbody></table>" +
        "<table border='1' style='width:100%;>" +
        "<tbody><tr><td colspan='5' style='text-align:center;><b>DETALLE DE REPARACIONES</b></td><td colspan='5' style='text-align:center;>" +
        "<td colspan='5' style='width:18%;><b>ID</b></td>" +
        "<td colspan='5' style='width:18%;><b>Fecha Completada</b></td>" +

```

```

" <td style="width:18%;><b>Fecha Completada</b></td> " +
" <td style="width:18%;><b>Pago</b></td> " +
" <td style="width:24%;><b>Entrega</b></td> " +
" <td style="width:18%;><b>Total</b></td></tr>" + detalleRetiro +
"</table></td><td style="width:2%;></td></tr></table></table> " +
"<br>" +
"<br><br><br>" +
"<footer><table style="width: 100%;><tr><td colspan="3"><hr></td></tr><tr>" +
" <td style="width: 20%;>" + fecha + "</td><td style="width: 60%; text-align: center;>Informe emitido por Sistema Web EPC-Compu</td>" +
" <td style="width: 20%;></td></tr></table></footer>" +
"</body> </html> ";

var pdf = Pdf
.From(html)
.WithGlobalSetting("orientation", "portrait")
.WithObjectSetting("web.defaultEncoding", "utf-8")
.Content();

Session["download1"] = pdf;
Response.Redirect("IDescargarEstado.aspx", true);
}

```

**Figura 207.** Método de generación de informe reparaciones completadas en la capa de interfaz  
**Elaborado por:** Valeria Quispe

## Tarea Completada



**Figura 208.** Gestión de Reportes completada  
**Elaborado por:** Valeria Quispe



**Figura 209.** Gestión de Reportes completada  
**Elaborado por:** Valeria Quispe

### **H13 Gestión de Usuarios**

La gestión de usuarios se enfoca en manejar la información de todos los usuarios que tienen acceso al sistema, con el fin de una mejor organización tanto del personal como de los clientes, en ello podrá realizar una visualización de usuarios, registro, actualización y baja.



**Figura 210.** Gestión de Usuarios  
**Elaborado por:** Valeria Quispe

### Prueba

La finalidad de la prueba es llevar un correcto funcionamiento en la gestión de usuarios, tanto así verificará la actividad de visualización, registro, actualización y baja de un registro.




**Figura 211.** Pruebas para la Gestión de Usuarios  
**Elaborado por:** Valeria Quispe











### **Evidencia**

La gestión de usuarios, presenta la visualización de todos los usuarios registrados en el sistema, este permite realizar la búsqueda para facilitar las operaciones de actualización y baja de un registro.

## Gestión de Usuarios

LISTADO DE USUARIOS 

Buscar:

Id Persona	Nombre	Apellido	Correo	Teléfono		
3	Valeria	Quispe	marco	0979398197		
4	Jessica	Pintag	jessica.pintag@gmail.com	0994864619		
5	David	Cervantes	david	0983741893		
6	David	Cervantes	Cervantes.78@gmail.com	0983741893		
7	Fernando	Naula	fernando	0979398197		


[Agregar Usuario](#)

**Figura 212.** Visualización de usuarios  
**Elaborado por:** Valeria Quispe

A su vez presenta el formulario para realizar el registro de los usuarios, donde presenta cajas para la recolección de la información personal y una vez digitado todos los datos se procederá a realizar el registro del usuario.

## Gestión de Usuarios

### Formulario de Registro

DATOS DE USUARIO 

<input type="text" value="Nombre de Usuario"/>	<input type="text" value="Apellido Usuario"/>
<input type="text" value="Administrador"/>	<input type="text" value="Correo Electrónico"/>
<input type="text" value="Teléfono de Usuario"/>	<input type="text" value="Dirección de Usuario"/>

Nota: La primer contraseña es su correo electrónico hasta antes del prefijo arroba (---@)

[Guardar](#) [Cancelar](#)

**Figura 213.** Formulario de registro de usuario  
**Elaborado por:** Valeria Quispe

La actualización de un usuario permite visualizar la información existente de un usuario, habilitando los campos que puedan actualizarse.

**Gestión de Usuarios**  
Actualizar Usuario

---

DATOS DE USUARIO 3 

Valeria	Quispe
Cliente	marco
0979398197	Ciudadela cumanda Alrededor

Nota: La contraseña puede ser actualizada por el propio usuario

Guardar Cancelar

**Figura 214.** Formulario de actualización de usuario  
**Elaborado por:** Valeria Quispe

### Codificación

Para recuperar el listado de usuarios se presenta el método `CargarListadoPerfilesUsuarios()`, mismo que hace el llamado al método de la capa de negocio `ObtenerUsuariosActivos()`.

```
private void cargarListadoPerfilesUsuarios()
{
    List<EUsuario> listausuarios = new List<EUsuario>();
    List<EPersona> listapersonas = new List<EPersona>();
    listausuarios = NUsuario.ObtenerUsuariosActivos();

    foreach (var item in listausuarios)
    {
        EPersona per = new EPersona();
        per = NPersona.DevolverPersonaxId(item.IdPersonaPertenece);
        listapersonas.Add(per);
    }
    GridViewUsuarios.DataSource = listapersonas;
    GridViewUsuarios.DataBind();
}
```

**Figura 215.** Método de recuperar usuarios activos en la capa de interfaz



**Elaborado por:** Valeria Quispe

Para realizar el registro de un usuario, primero se expone el método encargado de recuperar los perfiles existentes en el sistema, esto para facilitar la operación del usuario pertinente.

```
private void cargarListadoPerfiles()
{
    List<EPerfil> listadoPerfiles = new List<EPerfil>();
    listadoPerfiles = NUsuario.devolverListadoPerfilesNegocio();
    DropDownListPerfil.DataSource = listadoPerfiles;
    Session["Perfiles"] = listadoPerfiles;
    DropDownListPerfil.DataTextField = "NombrePerfil";
    DropDownListPerfil.DataValueField = "IdPerfil";
    DropDownListPerfil.DataBind();
}
```

**Figura 216.** Método de recuperar perfiles existentes en la capa de interfaz

**Elaborado por:** Valeria Quispe

Previo a realizar el registro de un usuario se contempla las validaciones necesarias, esto antes de llevar a cabo la operación y cuya finalidad es de mantener integridad en los datos almacenados.

```
public void RegistroUsuarioValidaciones()
{
    try
    {
        int correoExistente = NPersona.CorreoExistente(TextBoxCorreo.Text);
        if (correoExistente != 0)
        {
            LabelMensaje.Text = "Correo ya registrado";
        }
        else if (TextBoxNombreUsuario.Text == "" || TextBoxApellido.Text == "" ||
            TextBoxDireccion.Text == "" || TextBoxCorreo.Text == "" || TextBoxTelefono.Text == "")
        {
            LabelMensaje.Text = "Datos Incompletos";
        }
        else
        {
            registrarNuevoUsuario();
        }
    }
    catch (Exception)
    {
        LabelMensaje.Text = "Registro erróneo, intente más tarde";
    }
}
```

**Figura 217.** Método de validación de correo y datos en la capa de interfaz

**Elaborado por:** Valeria Quispe

Conforme a cumplir con todos los datos del usuario a guardar se lleva a cabo el método registrarNuevoUsuario, que recupera la información de las cajas de texto y lo transforma en entidades antes de enviarlo por los distintos métodos.

```
private void registrarNuevoUsuario()
{
    //Registrar Persona
    EPersona personaNueva = new EPersona();
    personaNueva.NombrePersona = TextBoxNombreUsuario.Text;
    personaNueva.ApellidoPersona = TextBoxApellido.Text;
    personaNueva.CorreoPersona = TextBoxCorreo.Text;
    personaNueva.TelefonoPersona = TextBoxTelefono.Text;
    personaNueva.DireccionPersona = TextBoxDireccion.Text;
    personaNueva = NPersona.GuardarPersonaNegocio(personaNueva);
    if (personaNueva == null)
    {
        LabelMensaje.Text = "Registro no completado";
    }
    else
    {
        //Registrar Usuario
        EPerfil perfil = new EPerfil();
        EUsuario nuevo = new EUsuario();
        perfil = NPersona.devolverPerfilxNombre(DropDownListPerfil.SelectedItem.ToString());
        string parametro = TextBoxCorreo.Text;
        string[] words = parametro.Split('@');
        string contra = words[0];
        nuevo.IdPerfilPertenece = perfil.IdPerfil;
        nuevo.ContraseniaUsuario = metodos.Encrypt(contra);
        nuevo.IdPersonaPertenece = personaNueva.IdPersona;
        nuevo.EstadoUsuario = 1;
        nuevo = NUsuario.GuardarUsuarioNegocio(nuevo);
        if (nuevo == null)
        {
            LabelMensaje.Text = "Registro no completado";
        }
        else
        {
            cargarListadoPerfilesUsuarios();
            MultiView1.ActiveViewIndex = 1;
        }
    }
}
```

**Figura 218.** Método de guardar usuario en la capa de interfaz  
**Elaborado por:** Valeria Quispe

Para cumplir con la actualización del usuario se desarrolla el método actualizarusuario(id), que recibe como parámetro el id de usuario a actualizar y recupera la información de interés de las cajas de texto perteneciente a la interfaz antes de realizar el llamado a los métodos para cumplir con la modificación de los datos del usuario.

```

private void actualizarusuario(string id)
{
    try
    {
        EPersona actualizar = new EPersona();
        actualizar.IdPersona = Convert.ToInt32(LabelIdUsuario.Text);
        actualizar.NombrePersona = TextBoxNombreUsuario.Text;
        actualizar.ApellidoPersona = TextBoxApellido.Text;
        actualizar.CorreoPersona = TextBoxCorreo.Text;
        actualizar.TelefonoPersona = TextBoxTelefono.Text;
        actualizar.DireccionPersona = TextBoxDireccion.Text;
        actualizar = NPersona.ActualizarPersona(actualizar);
        EUsuario actualizaru = new EUsuario();
        EUsuario n =NUsuario.DevolverUsuarioidPersona(actualizar.IdPersona);
        actualizaru.IdUsuario = n.IdUsuario;
        actualizaru.IdPersonaPertenece = actualizar.IdPersona;
        actualizaru.IdPerfilPertenece = retornarPerfil();
        actualizaru.EstadoUsuario = 1;
        actualizaru = NUsuario.ActualizarUsuario(actualizaru);
        if (actualizaru == null)
        {
            LabelMensaje.Text = "Registro no actualizado correctamente";
        }
        else
        {
            cargarListadoPerfilesUsuarios();
            limpiarCampos();
            //desbloquearCamposActualizar();
            LabelMensajeGrid.ForeColor = System.Drawing.Color.DarkGreen;
            LabelMensajeGrid.Text = "Registro Actualizado Correctamente";
            MultiView1.ActiveViewIndex = 1;
        }
    }
    catch (Exception)
    {
        LabelMensaje.ForeColor = System.Drawing.Color.DarkRed;
        LabelMensaje.Text = "Actualización errónea, intente más tarde";
    }
}

```

**Figura 219.** Método de actualizar usuario en la capa de interfaz  
**Elaborado por:** Valeria Quispe

Para finalizar con la baja de un registro se completa el código que recuperará el identificador del usuario de interés, con el fin de actualizar el estado del estado del usuario ya que no se puede eliminar directamente un registro esto para mantener la integridad de los datos y también que ya posee relaciones con las diferentes entidades.

```

protected void GridViewUsuarios_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    try
    {
        int idPersona = Convert.ToInt32(GridViewUsuarios.DataKeys[e.RowIndex].Values[0]);
        EUsuario eliminar = NUsuario.DevolverUsuarioxIdPersona(idPersona);
        int borrado = NUsuario.EliminarUsuario(eliminar.IdUsuario);
        if (borrado == 1)
        {
            cargarListadoPerfilesUsuarios();
            LabelMensajeGrid.ForeColor = System.Drawing.Color.DarkBlue;
            LabelMensajeGrid.Text = "Registro Dado de Baja exitosamente";
        }
        else if (borrado == 0)
        {
            LabelMensajeGrid.ForeColor = System.Drawing.Color.DarkRed;
            LabelMensajeGrid.Text = "Operación errónea, Comuníquese con el Administrador";
        }
    }
    catch (Exception)
    {
        LabelMensajeGrid.Text = "Baja errónea, intente más tarde";
    }
}

```

**Figura 220.** Método de dar baja un usuario en la capa de interfaz  
**Elaborado por:** Valeria Quispe

### Tarea Completa



**Figura 221.** Gestión de Usuarios completada  
**Elaborado por:** Valeria Quispe



**Figura 222.** Gestión de Usuarios completada  
**Elaborado por:** Valeria Quispe

#### **H14 Acceso al menú por tipo de usuario**

La historia de usuario contempla una correcta habilitación de las operaciones que tendrá asignado cada usuario como: secretaria, cliente, personal de entrega y recepción, personal de soporte técnico y el administrador todo conforme a llevar un correcto manejo del servicio ofrecido por EPC-COMPU.



**Figura 223.** Acceso al Menú por tipo de usuario  
**Elaborado por:** Valeria Quispe

### Prueba

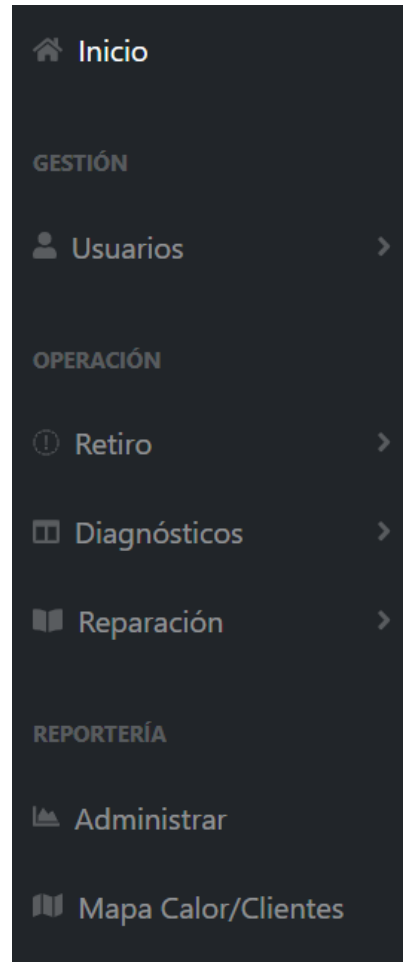
La finalidad de las pruebas es ingresar al sistema con cada perfil de usuario existente y verificar la habilitación de las operaciones conforme a las diferentes gestiones que pone a disposición el sistema web.



**Figura 224.** Pruebas para el Acceso al menú por tipo de usuario  
**Elaborado por:** Valeria Quispe

### **Evidencia**

Se presenta las operaciones de las diferentes gestiones que conforma el Sistema Web EPC- COMPU



**Figura 225.** Menú de operaciones  
**Elaborado por:** Valeria Quispe

### **Codificación**

Se presenta un apartado de la habilitación de las diferentes operaciones, esto facilitando la administración de los diferentes usuarios.



```

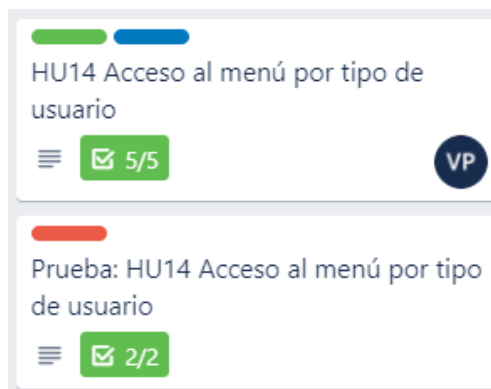
window.onload=function () {
    var perfil=<%=idperfil()%>;

    if (perfil == 1) { //Administrador
        document.getElementById('menuRetiro').style.display = 'none';
        document.getElementById('menuDiagnostico').style.display = 'none';
        document.getElementById('menuReparacion').style.display = 'none';
    } else if (perfil == 2) { //Cliente
        document.getElementById('menuAdministrador').style.display = 'none';
        document.getElementById('menuInformes').style.display = 'none';
    } else if (perfil == 3) { //Secretaria
        document.getElementById('menuAdministrador').style.display = 'none';
        document.getElementById('menuDiagnostico').style.display = 'none';
    } else if (perfil == 4) { //Tecnico
        document.getElementById('menuAdministrador').style.display = 'none';
        document.getElementById('menuRetiro').style.display = 'none';
    } else if (perfil == 5) { //Personal de E/R
        document.getElementById('menuAdministrador').style.display = 'none';
        document.getElementById('menuDiagnostico').style.display = 'none';
        document.getElementById('menuReparacion').style.display = 'none';
    }
}
}

```

**Figura 226.** Habilitación de menús por perfil de usuario  
**Elaborado por:** Valeria Quispe

## Tarea Completada



**Figura 227.** Acceso al menú por tipo de usuario completada  
**Elaborado por:** Valeria Quispe



**Figura 228.** Acceso al menú por tipo de usuario completada  
**Elaborado por:** Valeria Quispe

### **H15 Manejo de Perfil Personal de los Usuarios**

El manejo del perfil personal permite al usuario visualizar la información personal de interés y a su vez permite realizar la modificación de los datos habilitados por el sistema web.



**Figura 229.** Manejo de Perfil Personal de los Usuarios  
**Elaborado por:** Valeria Quispe

### Prueba

La finalidad de la prueba es evaluar la correcta presentación de la información personal del usuario de ingreso y así como la actualización de los datos permitidos.



**Figura 230.** Pruebas del Manejo de Perfil Personal de los Usuarios  
**Elaborado por:** Valeria Quispe


### **Evidencia**

Se expone la interfaz de presentación de información personal del usuario de ingreso, esto para ayudar en la modificación de datos de interés.

## Perfil de Usuario

Información

---



DATOS DE USUARIO

Valeria	Quispe
Cliente	marco
0979398197	Ciudadela cumanda Alrededor

[Modificar Información](#)

**Figura 231.** Vista de perfil de usuario  
**Elaborado por:** Valeria Quispe

A su vez se presenta el formulario de actualización de información del usuario, para que pueda modificar los datos de interés y los que se encuentren habilitados.

## Perfil de Usuario

Actualizar Información

---

DATOS DE USUARIO 3 

<input type="text" value="Valeria"/>	<input type="text" value="Quispe"/>
<input type="text" value="Cliente"/>	<input type="text" value="marco"/>
<input type="text" value="0979398197"/>	<input type="text" value="Ciudadela cumanda Alrededor"/>
<input type="text" value="BYYw6GNAShf386+EfkVWQg=="/>	<input type="text" value="BYYw6GNAShf386+EfkVWQg=="/>
<input type="checkbox"/> ¿Sí, Desea cambiar contraseña seleccione el casillero?	

**Figura 232.** Formulario de actualización de perfil de usuario  
**Elaborado por:** Valeria Quispe

### Codificación

Conforme a cumplir con el desarrollo de la funcionalidad de manejo de perfil personal de los usuarios, se contempla la realización del método encargado de recuperación la información personal del usuario para ser presentado en la interfaz de usuario.

```

private void cargarUsuario(EPersona per, EUsuario user)
{
    Session["PersonaA"] = per;
    Session["UsuarioA"] = user;

    TextBoxApellido.Text = per.ApellidoPersona;
    TextBoxCorreo.Text = per.CorreoPersona;
    TextBoxDireccion.Text = per.DireccionPersona;
    TextBoxNombreUsuario.Text = per.NombrePersona;
    TextBoxTelefono.Text = per.TelefonoPersona;
    int perfil = user.IdPerfilPertenece;
    if (perfil == 1)
    {
        DropDownListPerfil.SelectedIndex = 0;
    }
    else if (perfil == 2)
    {
        DropDownListPerfil.SelectedIndex = 1;
    }
    else if (perfil == 3)
    {
        DropDownListPerfil.SelectedIndex = 2;
    }
    else if (perfil == 4)
    {
        DropDownListPerfil.SelectedIndex = 3;
    }
    else if (perfil == 5)
    {
        DropDownListPerfil.SelectedIndex = 4;
    }
}

```

**Figura 233.** Método de cargar información de usuario en la capa de interfaz

**Elaborado por:** Valeria Quispe

A su vez se lleva a cabo el método encargado de realizar la actualización de la información de los datos del usuario.

```

private void actualizarusuario(string id)
{
    try
    {
        EPersona actualizar = new EPersona();
        actualizar.IdPersona = Convert.ToInt32(Label2.Text);
        actualizar.NombrePersona = TextBoxNombreA.Text;
        actualizar.ApellidoPersona = TextBoxApellidoA.Text;
        actualizar.CorreoPersona = TextBoxCorreoA.Text;
        actualizar.TelefonoPersona = TextBoxTelefonoA.Text;
        actualizar.DireccionPersona = TextBoxDireccionA.Text;
        actualizar = NPersona.ActualizarPersona(actualizar);

        EUsuario actualizaru = new EUsuario();
        EUsuario n = NUusuario.DevolverUsuarioIdPersona(actualizar.IdPersona);
        actualizaru.IdUsuario = n.IdUsuario;
        actualizaru.IdPersonaPertenece = actualizar.IdPersona;

        if (CheckBoxCambiarContrasenia.Checked)
        {
            if (TextBoxContraseniaA.Text == TextBoxRepetirContraseniaA.Text)
            {
                actualizaru.ContraseniaUsuario = metodos.Encrypt(TextBoxContraseniaA.Text);
            }
            else
            {
                Label3.Text = "Contraseñas no coinciden";
            }
        }
        else
        {
            actualizaru.ContraseniaUsuario = metodos.Encrypt(metodos.Decrypt(n.ContraseniaUsuario));
        }

        if (actualizaru == null)
        {
            Label3.Text = "Perfil no actualizado revise, información";
        }
        else
        {
            limpiarCampos();
            LabelMensaje.ForeColor = System.Drawing.Color.DarkBlue;
            LabelMensaje.Text = "Perfil Actualizado Correctamente";
            CheckBoxCambiarContrasenia.Checked = false;
            cargarUsuario(actualizar, actualizaru);
            Label3.Text = "";
            MultiView1.ActiveViewIndex = 0;
        }
    }
    catch (Exception)
    {
        Label3.Text = "Actualización errónea, intente más tarde";
    }
}

```

**Figura 234.** Método de actualizar usuario en la capa de interfaz  
**Elaborado por:** Valeria Quispe

**Tarea Completada**





**Figura 235.** Manejo de Perfil Personal de los Usuarios completada

**Elaborado por:** Valeria Quispe



**Figura 236.** Manejo de Perfil Personal de los Usuarios completada

**Elaborado por:** Valeria Quispe

### **3.3.7 Retroalimentación**

## **CAPÍTULO IV.- CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

- Mediante la información obtenida a través de las encuestas y la entrevista al personal de EPC Compu, se ha logrado identificar el proceso completo que maneja la empresa en la prestación del servicio de Soporte Técnico, siendo fundamental en la consecución de los requerimientos del sistema web y con la información recolectada ayudó a la elaboración de las historias de usuario para su previo desarrollo
- A través del levantamiento de requerimientos, se obtuvo una parte del proceso en el cual el cliente solicita la entrega y recepción de dispositivos a domicilio por lo cual se lo implemento utilizando georreferenciación, obteniendo mejores resultados en la organización de la empresa y la fácil interacción de los clientes con el sistema al no tener que solicitar el servicio adicional mediante otros medios.
- Para la elección tanto del lenguaje y Framework idóneo para el proyecto, como se pudo notar se lo realizó a través de cuadros comparativos los cuales ayudaron a elegir la mejor opción acorde con los requerimientos del cliente y con las capacidades del investigador.
- El uso de .NET Framework permite la integración de herramientas de gran ayuda durante el proceso de desarrollo como lo es Visual Studio que al ser un IDE completo se lo utilizo para las distintas tareas dentro del proyecto como la edición, depuración y compilación para su posterior publicación.

- Se logro un mejor desempeño y organización durante la elaboración del proyecto debido a que se estableció el uso de una metodología de desarrollo de software, la cual permitió que se realicen cambios durante las distintas pruebas del software sin mayor inconveniente durante el tiempo de culminación de cada iteración.
- Implantar el sistema web para la gestión de soporte técnico ayudó de manera positiva a la organización del equipo de trabajo en la empresa, debido a que se tuvo un mayor control de todos los soportes técnicos realizados y no realizados, además de tener una ventaja competitiva como es la entrega y recepción de dispositivos a domicilio aplicando georreferenciación de tal manera que el personal encargado tenga una mejor idea del lugar a cuál debe ir.

#### **4.2 Recomendaciones**

- Se sugiere mantener el sitio web alojado en el proveedor de confianza asignado durante la implantación, debido a que se puede incluir certificados y manejarlos dentro del mismo de manera fácil y sencilla, para así evitar contratar proveedores externos que no sean seguros y deban realizar una nueva investigación.
- Como recomendación, se debe realizar copias de seguridad de la página web y de la base de datos periódicamente, y así debido a cualquier tipo de imprevisto, se pueda realizar la recuperación de una versión actualizada y evitar hacer un trabajo desde cero.
- Se recomienda que a lo largo del tiempo mediante el uso del sistema web se lo vaya implementando nuevos módulos de acuerdo vayan generando requerimientos, en los cuales ayude a mejorar cada día la organización y el desempeño de la empresa EPC Compu.

- Como recomendación, se le indica a la empresa EPC Compu que el api de Google Maps dispone de peticiones limitadas por día por lo cual según se vaya acoplado el sistema a los clientes lo puedan contratar en un plan de acuerdo con el número de peticiones necesarias.
- Se recomienda depurar información innecesaria dentro de la base de datos para evitar inconsistencias en el sistema.
- Se recomienda al desarrollador o persona encargada del sistema, que inicie por el estudio de .NET debido a que la curva de aprendizaje es óptima, para que pueda realizar modificaciones o ampliación de módulos dentro del sistema.

## BIBLIOGRAFÍA

- [1] J. R. Maldonado Rodrigo, “DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA WEB DE SEGUIMIENTO Y EVALUACIÓN DE LAS PRÁCTICAS PRE-PROFESIONALES PARA LA FACULTAD DE INGENIERÍA ESCUELA CIVIL DE LA PUCE”, p. 198, 2018, Consultado: jul. 03, 2022. [En línea]. Disponible en: [http://repositorio.puce.edu.ec/bitstream/handle/22000/12562/Tesis\\_Teoria.pdf?sequence=1&isAllowed=y](http://repositorio.puce.edu.ec/bitstream/handle/22000/12562/Tesis_Teoria.pdf?sequence=1&isAllowed=y).
- [2] T. Rodríguez, R. Dos Santos, y J. Aguilar, “Metodología para el Desarrollo de Aplicaciones Web Utilizando Datos Enlazados”, en *Conferencia Nacional de Computación, Informática y Sistemas (CoNCISa 2017)*, 2017, p. 9, Consultado: jul. 03, 2022. [En línea]. Disponible en: <https://infolibros.org/pdfview/11893-metodologia-para-el-desarrollo-de-aplicaciones-web-utilizando-datos-enlazados-articulo-taniana-rodriguez-ricardo-dos-santos-y-jose-aguilar/>.
- [3] V. O. EDISON FERNANDO, “DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN MONOGRAFÍA: PREVIO A LA OBTENCIÓN DEL”, p. 85, 2017, Consultado: jul. 03, 2022. [En línea]. Disponible en: <http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/13776/M-ESPE-057503.pdf?sequence=1&isAllowed=y>.
- [4] L. D. MANCHENO SEGOVIA, “Aplicación Web Para La Gestión De Información Del Departamento Técnico De Reparación Y Garantías De La Empresa Telnets Soporte De La Ciudad De Ibarra”, vol. 53, núm. 9, pp. 1689–1699, 2018, Consultado: jul. 03, 2022. [En línea]. Disponible en: <https://dspace.uniandes.edu.ec/bitstream/123456789/9601/1/TUAEXCOMSIS003-2019.pdf>.
- [5] Destino Negocio, “Las barreras de las pymes para subirse a la tecnología”, *Destino Negocio*, 2016. <https://negociosdigitalesmovistar.com/emprendimiento/las-barreras-de-las-pymes-para-subirse-a-la-tecnologia/> (consultado jul. 03, 2022).
- [6] G. Coba, “Nueve de cada 10 pymes en Ecuador invierte en tecnología”, *Primicias*, 2022, Consultado: jul. 03, 2022. [En línea]. Disponible en: <https://www.primicias.ec/noticias/economia/pymes-ecuador-inversion-tecnologia-pandemia/>.
- [7] A. G. Altamirano Velastegui y W. M. Freire Pico, “UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD DE CIENCIAS ADMINISTRATIVAS”, p. 136, 2019.
- [8] Universidad ETAC, “¿Qué es TI?: conoce este concepto y sus alcances”, 2020. <https://etac.edu.mx/blog-etac/index.php/que-es-ti/> (consultado jul. 03, 2022).
- [9] M. Quirós Hernández, “Tecnologías de la Información Geográfica (TIG). Cartografía, Fotointerpretación, Teledetección y SIG | Ediciones Universidad de Salamanca”, 2011, 2011. <https://eusal.es/index.php/eusal/catalog/book/978-84-7800-148-4> (consultado jul. 03, 2022).
- [10] Aeroterra, “¿Qué es SIG? | Sistemas de Información Geográfica - Aeroterra

- S.A.” <https://www.aeroterra.com/es-ar/que-es-gis/introduccion> (consultado jul. 03, 2022).
- [11] J. Palacios, *Topografía para las tropas*. 2005.
- [12] A. S. Aldáz, “Aplicación móvil de geolocalización para gestión y control de vendedores de la empresa ‘SKANDINAR S.A.’”, 2021, Consultado: jul. 08, 2022. [En línea]. Disponible en: <https://repositorio.uta.edu.ec/bitstream/123456789/33488/1/t1866si.pdf>.
- [13] Raquel Maluenda, “Tipos de desarrollo de aplicaciones web: ejemplos y características”, *Tipos de desarrollo de aplicaciones web: ejemplos y características*, 2020. <https://profile.es/blog/desarrollo-aplicaciones-web/> (consultado jul. 08, 2022).
- [14] IngenioVirtual, “¿Qué es una aplicación web y cuáles son sus ventajas? - ingeniovirtual.com”. <https://www.ingeniovirtual.com/que-es-una-aplicacion-web-y-cuales-son-sus-ventajas/> (consultado jul. 08, 2022).
- [15] euskadi.eus, “Diseño adaptable (Responsive Design) - Servicio Web - Euskadi.eus”, 2020. <https://www.euskadi.eus/disenio-adaptable-responsive-design/web01-a2wz/es/> (consultado jul. 08, 2022).
- [16] D. Bernal González, “Tipos de aplicaciones móviles: ventajas, desventajas y ejemplos”, *Profile*, p. 5, 2021, Consultado: jul. 08, 2022. [En línea]. Disponible en: <https://profile.es/blog/tipos-aplicaciones-moviles-ventajas-ejemplos/>.
- [17] “1.2 Arquitectura de las aplicaciones web | mi-pagina-web”. <https://nagaaralive0012rh.wixsite.com/mi-pagina-web/12-arquitectura-de-las-aplicaciones-web> (consultado jul. 08, 2022).
- [18] Microsoft Corporation, “What is .NET? An open-source developer platform.”, 2021. <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet> (consultado jul. 08, 2022).
- [19] Miguel B, “.NET: Qué es y cómo funciona | Conoce el framework de Microsoft”, 2018. <https://www.cursosaula21.com/que-es-net/> (consultado jul. 08, 2022).
- [20] Rafael Altube Vera, “Qué es Laravel: Características y ventajas | OpenWebinars”, *openwebinars*, 2021. <https://openwebinars.net/blog/que-es-laravel-caracteristicas-y-ventajas/> (consultado jul. 08, 2022).
- [21] J. E. Jurado, “IMPLEMENTACIÓN DE UN SISTEMA WEB DE ATENCIÓN MÉDICA UTILIZANDO YII FRAMEWORK EN EL CONSULTORIO MÉDICO GUTIÉRREZ”, 2022.
- [22] SERGIO DARIAS PÉREZ, “¿QUÉ ES MICROSOFT SQL SERVER Y PARA QUÉ SIRVE?”, 18 OCTUBRE, 2021. <https://intelequia.com/blog/post/2948/que-es-microsoft-sql-server-y-para-que-sirve> (consultado jul. 08, 2022).
- [23] Universitad carlemany, “Metodologías de desarrollo de software | Universitat Carlemany”. <https://www.universitatcarlemany.com/actualidad/metodologias-de-desarrollo-de-software> (consultado jul. 08, 2022).
- [24] Web Admin, “5 beneficios de aplicar metodologías ágiles en el desarrollo de

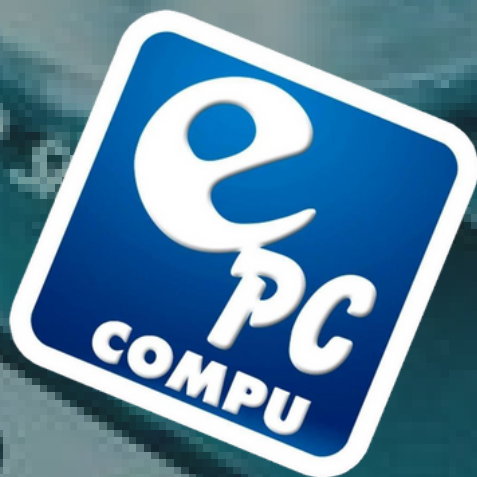
- software”, *Consultec*, 2017. <https://www.consultec-ti.com/b/5-beneficios-de-aplicar-metodologias-agiles-en-el-desarrollo-de-software> (consultado jul. 08, 2022).
- [25] santander universidades, “Metodologías de desarrollo software | Blog Becas Santander”, *becas-santander*, 2020. <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html> (consultado jul. 08, 2022).
- [26] A. Robledano, “Qué es NET Framework | OpenWebinars”, 2019. <https://openwebinars.net/blog/que-es-net-framework/> (consultado jul. 08, 2022).
- [27] Meta Platforms Inc., “React – Una biblioteca de JavaScript para construir interfaces de usuario”, *Una biblioteca de JavaScript para construir interfaces de usuario*, 2022. <https://es.reactjs.org/> (consultado jul. 08, 2022).
- [28] M. Ramis, “¿Qué es Angular y para qué sirve?”, *Quality devs*, 2019. <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/> (consultado jul. 08, 2022).
- [29] epitech, ? “Qué es PHP y para qué sirve? - Epitech España”, 2021. <https://www.epitech-it.es/que-es-php/> (consultado jul. 08, 2022).
- [30] M. Presta, “Las 10 mejores tecnologías de backend”. <https://blog.back4app.com/es/las-10-mejores-tecnologias-de-backend/#Python> (consultado jul. 08, 2022).
- [31] redusers, “ASP.NET: qué es y para qué sirve - RedUSERS”. <https://www.redusers.com/noticias/publicaciones/net/> (consultado jul. 08, 2022).
- [32] Deloitte, “¿Cuál es la metodología más adecuada para tu proyecto?”, *Deloitte.com*, 2018. <https://www2.deloitte.com/es/es/pages/technology/articles/waterfall-vs-agile.html> (consultado jul. 13, 2022).
- [33] C. Drumond, “Scrum: qué es, cómo funciona y por qué es excelente”, *Atlassian*. 2022, Consultado: jul. 08, 2022. [En línea]. Disponible en: <https://www.atlassian.com/es/agile/scrum>.
- [34] Google, “Planes de precios y costos de API: Google Maps Platform”. <https://mapsplatform.google.com/pricing/?hl=es-419> (consultado jul. 08, 2022).

## ANEXOS

### Anexo 1 : Manual de Usuario

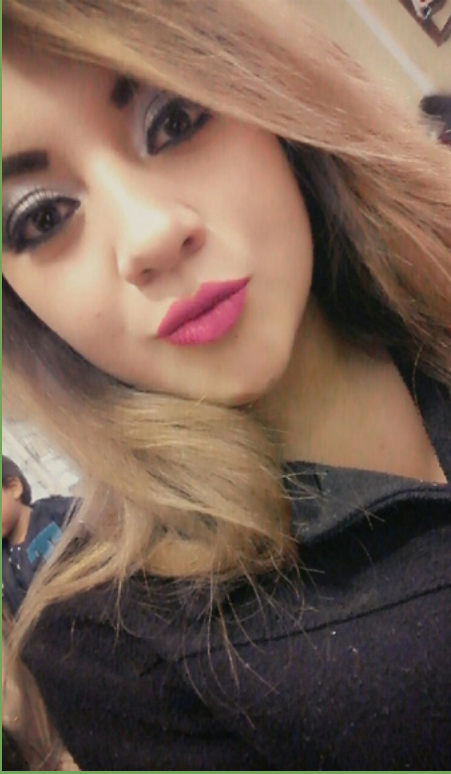
# MANUAL DE USUARIO

SISTEMA WEB





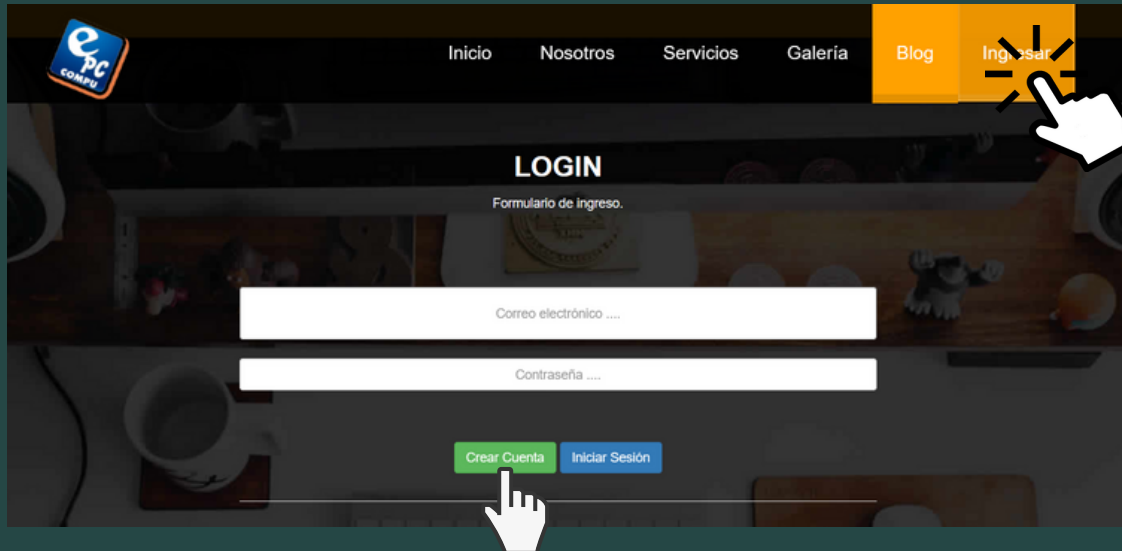
# Autora



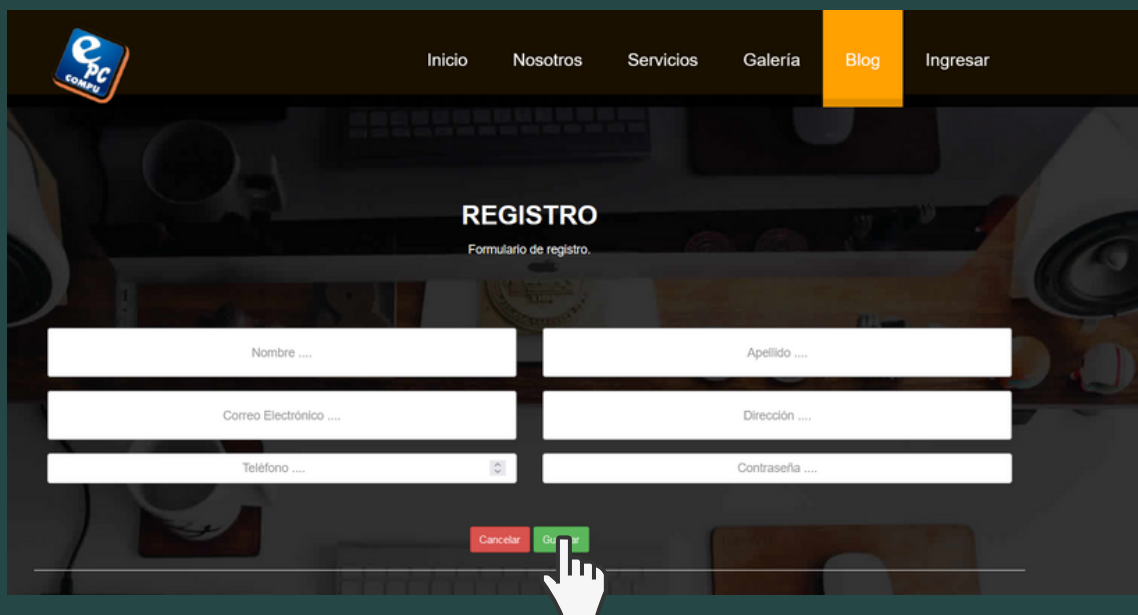
Valeria Quispe  
Faculta de Ingenieria en  
Sistemas, Electronica e  
Industrial  
Universidad Tecnica de  
Ambato

# Crear una cuenta

1. Ingresamos a la pagina web
2. Se dirige al apartado Ingresar
3. Presiona el boton Crear Cuenta

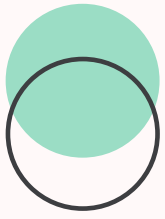


4. Llena los datos del formulario y presiona en Guardar



Su usuario sera el correo electronico ingresado en el formulario

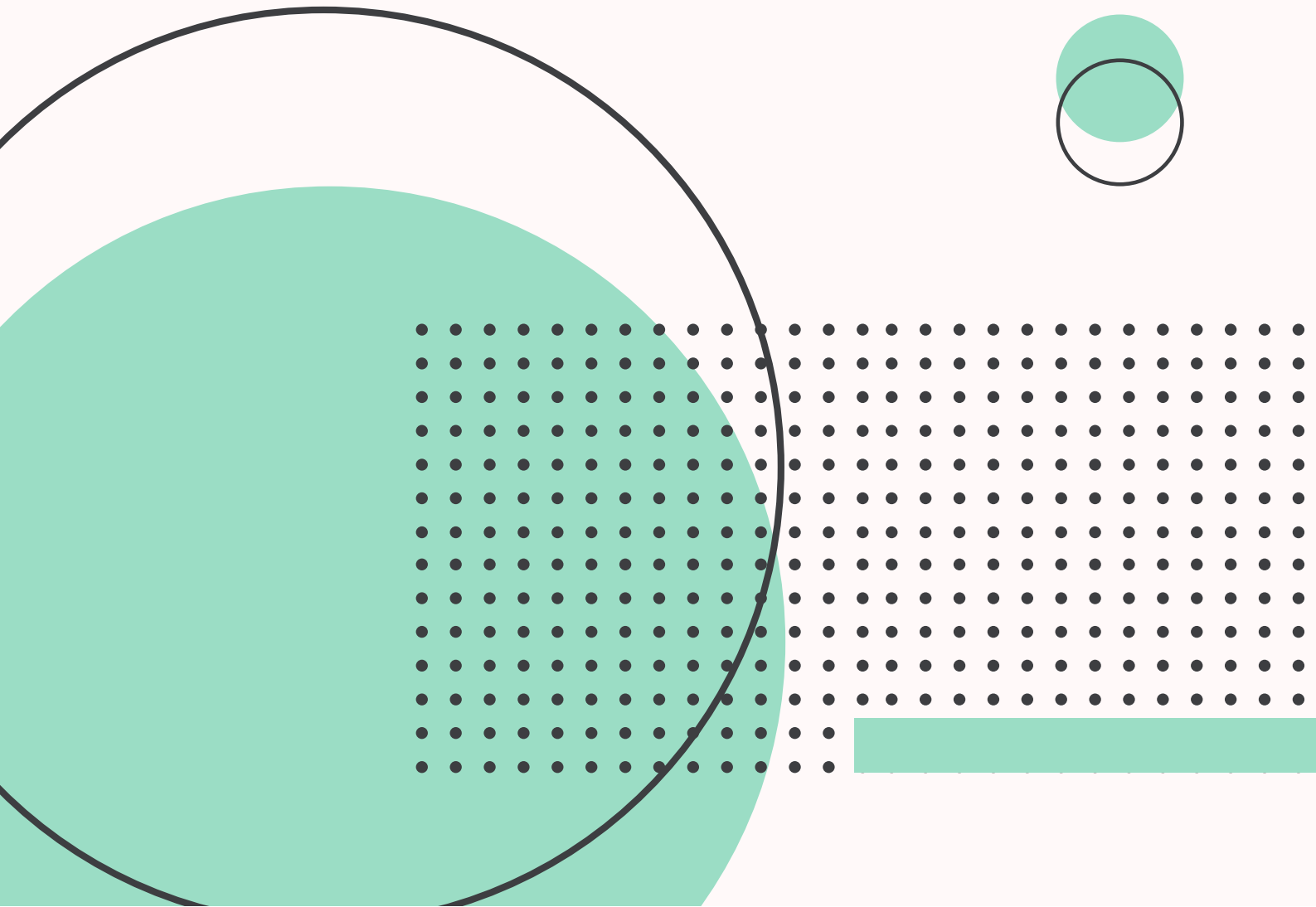
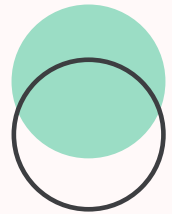
2022



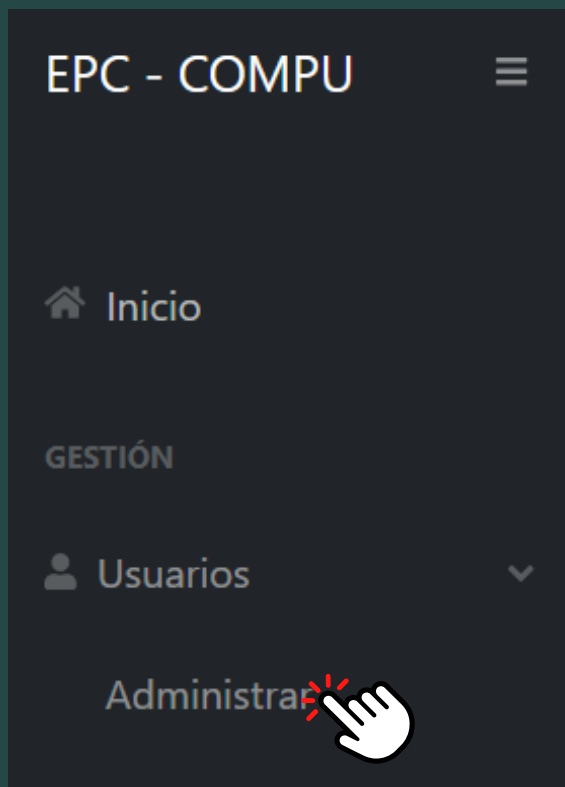
# ADMINISTRADOR

---

EPC COMPU



1. Ingresa las credenciales de administrador
2. Desplega el menu Usuarios
3. Selecciona Administrar



4. Aparece la lista de todos los usuarios registrados en el sistema

Gestión de Usuarios

LISTADO DE USUARIOS

Buscar:

Id Persona	Nombre	Apellido	Correo	Teléfono		
3	Valeria	Quispe	marco	0979398197		
4	Jessica	Pintag	jessica.pintag@gmail.com	0994864619		
5	David	Cervantes	david	0983741893		
7	Fernando	Naula	fernando	0979398197		
1007	Jhofre	Sinchi	jhofre.pintag@gmail.com	0979398197		
1008	Andrea	Quispe	andrea.quispe1097@gmail.com	0959019583		
1009	Sandra	Pazmiño	epccompu@gmail.com	0989274850		
1010	Mauricio	Ramos	mauri21072000@gmail.com	0986289360		
1011	Lisbeth	Narvaez	tefi_prins@hotmail.es	0962561279		
1012	Tiffani	Saltos	tiffanysaltos12@gmail.com	0999821718		

Agregar Usuario

Editar

Eliminar

Agregar Usuario

# Editar Usuario

1. Presiona el icono editar, y aparecen todos los datos que puede editar del usuario



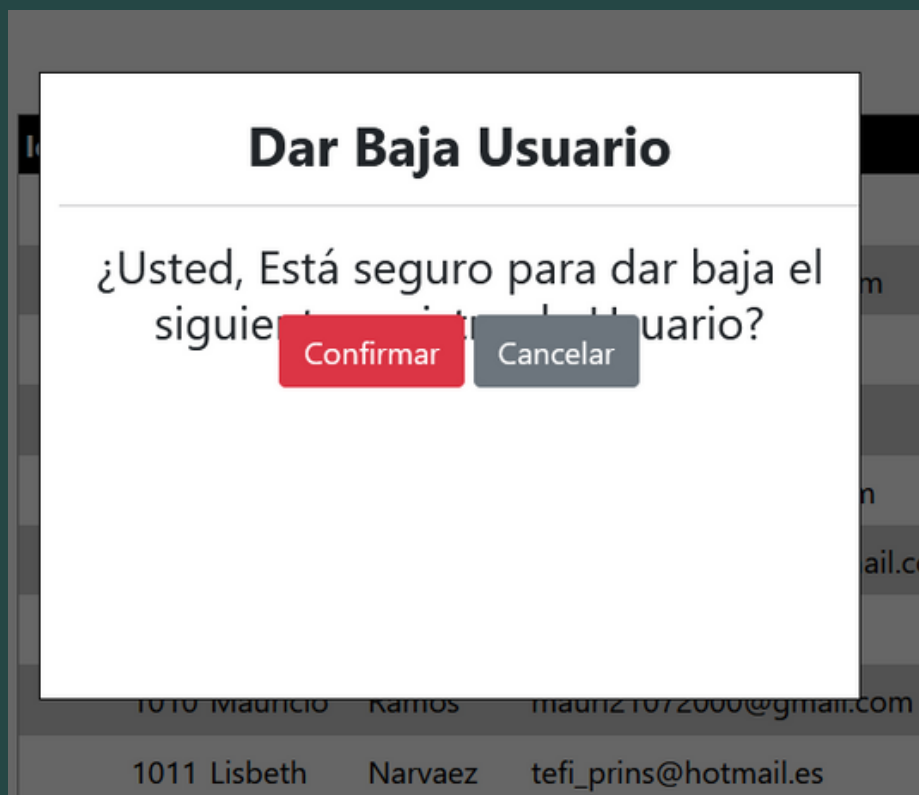
The screenshot shows a web application interface for user management. The header includes 'EPC - COMPU' on the left and 'David Cervantes' on the right. A dark sidebar on the left contains navigation items: 'Inicio', 'GESTIÓN', 'Usuarios', 'OPERACIÓN', 'REPORTERÍA', 'Administrar', and 'Mapa Calor/Clientes'. The main content area is titled 'Gestión de Usuarios' and 'Actualizar Usuario'. It features a 'DATOS DE USUARIO' section with a user icon. Below this are six input fields: 'Valeria', 'Quispe', 'Cliente', 'marco', '0979398197', and 'Ciudadela cumanda'. A note states: 'Nota: La contraseña puede ser actualizada por el propio usuario'. At the bottom are 'Guardar' and 'Cancelar' buttons.

2. Una vez actualizados los datos, presiona el boton Guardar

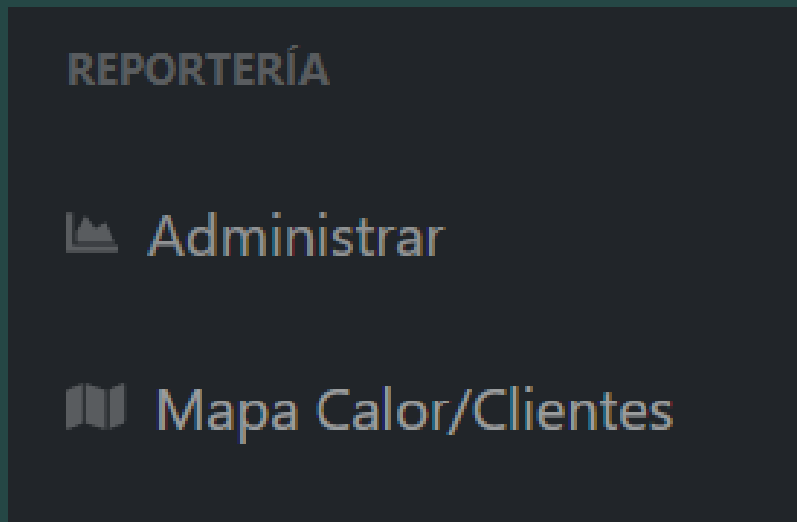
El correo y contraseña podra cambiar unicamente el usuario

# Eliminar Usuario

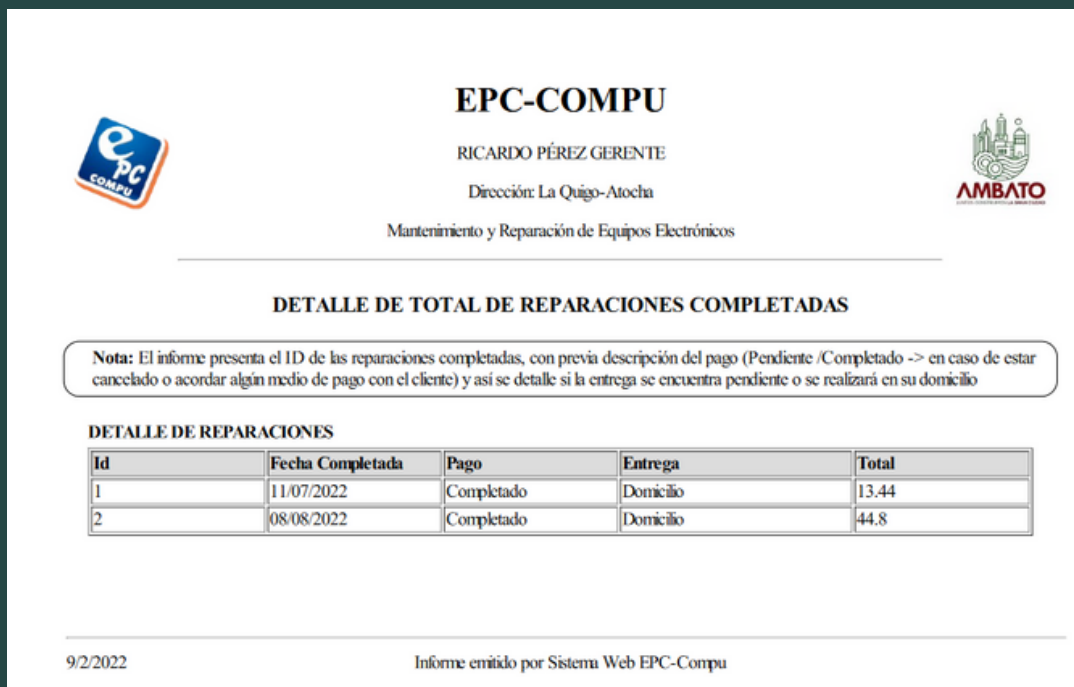
1. Presiona el icono eliminar, y aparecera un aviso en el cual te preguntara si esta de acuerdo en dar de baja dicho usuario.





Dentro del menu de reportería se puede visualizar dos opciones que son las siguientes



Administrar: Reportes de todas las reparaciones completadas



**EPC-COMPU**  
RICARDO PÉREZ GERENTE  
Dirección: La Quigo-Atocha  
Mantenimiento y Reparación de Equipos Electrónicos



**DETALLE DE TOTAL DE REPARACIONES COMPLETADAS**

**Nota:** El informe presenta el ID de las reparaciones completadas, con previa descripción del pago (Pendiente /Completado -> en caso de estar cancelado o acordar algún medio de pago con el cliente) y así se detalle si la entrega se encuentra pendiente o se realizará en su domicilio

**DETALLE DE REPARACIONES**

Id	Fecha Completada	Pago	Entrega	Total
1	11/07/2022	Completado	Domicilio	13.44
2	08/08/2022	Completado	Domicilio	44.8

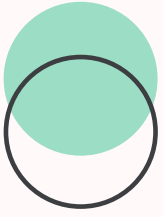
9/2/2022 Informe emitido por Sistema Web EPC-Compu

Mapa de Calor/Clientes: Nos mostrara un mapa con las diferentes ubicaciones de todos los clientes para tener conocimiento de donde estan la mayoría de ellos

The screenshot shows a web application interface with a dark sidebar on the left and a main content area. The sidebar contains the following menu items: Inicio, GESTIÓN, Usuarios, OPERACIÓN, REPORTERÍA, Administrar, and Mapa Calor/Clientes. The main content area has a header 'EPC - COMPU' and a user name 'David Cervantes'. The main title is 'Visualización de Mapa de Calor'. Below the title is a box labeled 'INFORMACIÓN ADICIONAL' containing the text 'Total de Clientes : 7' and a person icon. The main part of the interface is a satellite map with several green location markers. The map includes a 'Mapa' / 'Satélite' toggle, a full-screen icon, a person icon, and zoom controls. The footer of the map area includes the Google logo and copyright information: 'Combinaciones de teclas | Imágenes ©2022 TerraMetrics | Términos de uso'.



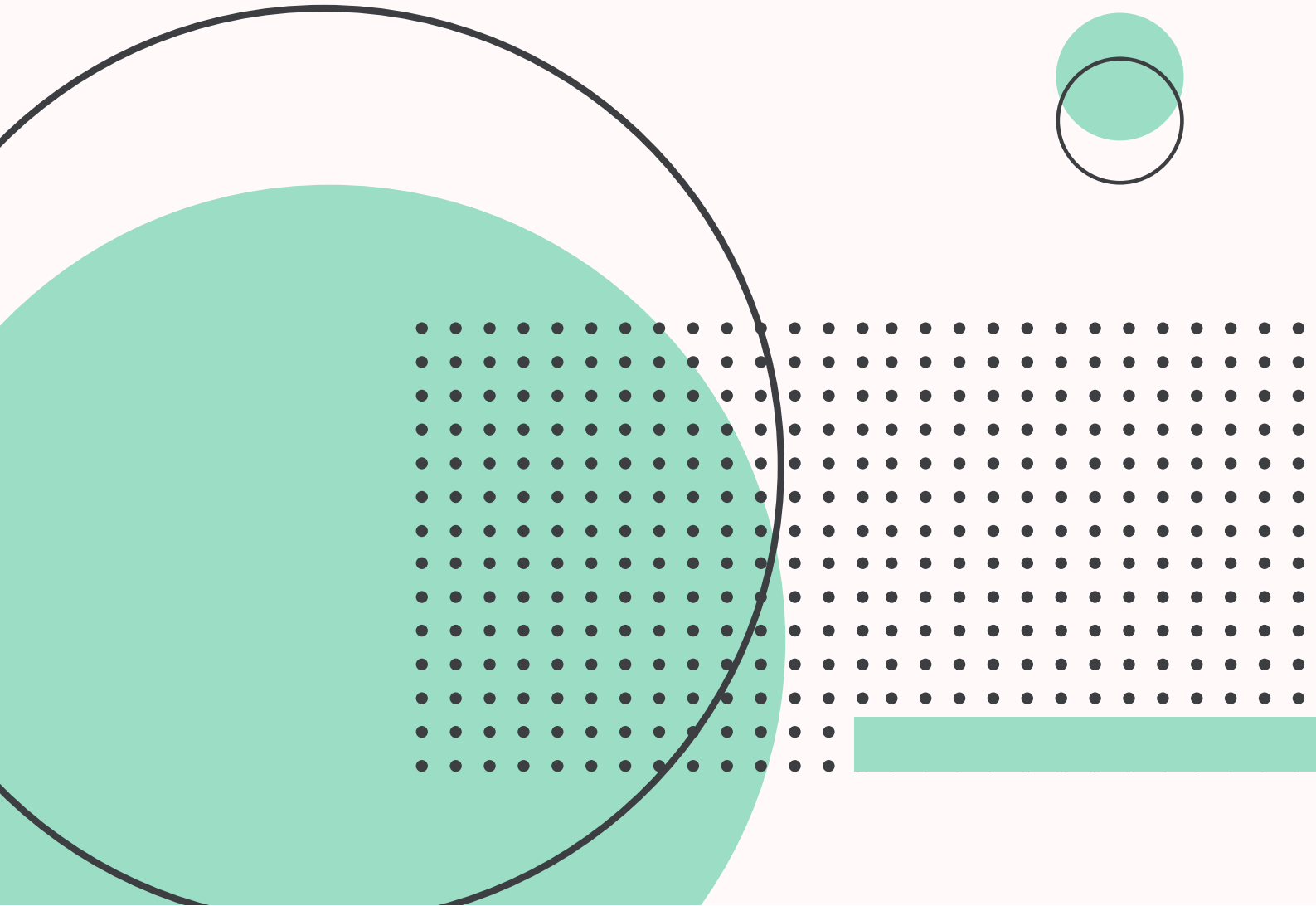
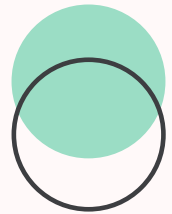
2022



# SECRETARIA

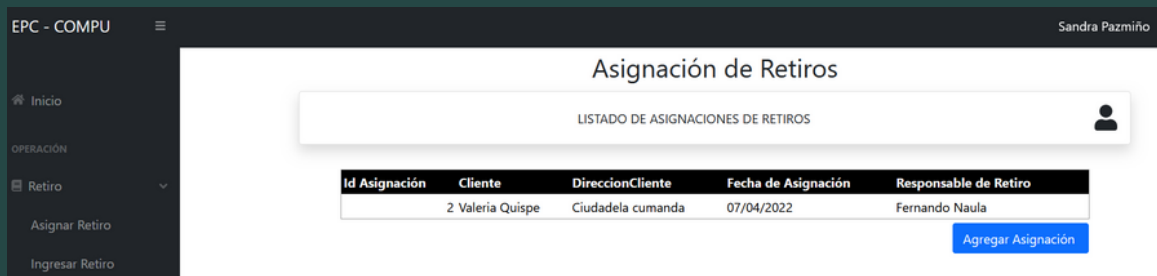
---

EPC COMPU



1. Ingresa las credenciales de secretaría
2. Desplegar el menu Retiro en el cual tiene dos opciones

# Asignar Retiros



Presiona el boton de Agregar Asignación y se desplegara una ventana



Ingresa el retiro y el personal que se hara cargo. Para ello presiona en el boton Buscar y aparecera el listado de retiro Pendientes, selecciona uno de ellos



## Listado de Retiros Pendientes

Id Retiro	Cliente	Fecha Solicitada	Día-Hora	Ubicación	Seleccionar
2	Valeria Quispe	29/03/2022	Miercoles 16:00	Ciudadela cumanda	
4	Valeria Quispe	30/03/2022	Miercoles 16:00	Ciudadela cumanda	
5	Jhofre Sinchi	05/08/2022	Miercoles 16:00	Vista hermosa	
15	Jhofre Sinchi	05/08/2022	Miercoles 15:00	Vista hermosa	
18	Jhofre Sinchi	06/08/2022	Miercoles 15:00	Vista hermosa	
19	Jhofre Sinchi	06/08/2022	Jueves 15:00	Vista hermosa	
20	Jhofre Sinchi	08/08/2022	Jueves 16:00	Vista hermosa	

Cerrar

De la misma manera lo hace para seleccionar el personal encargado.

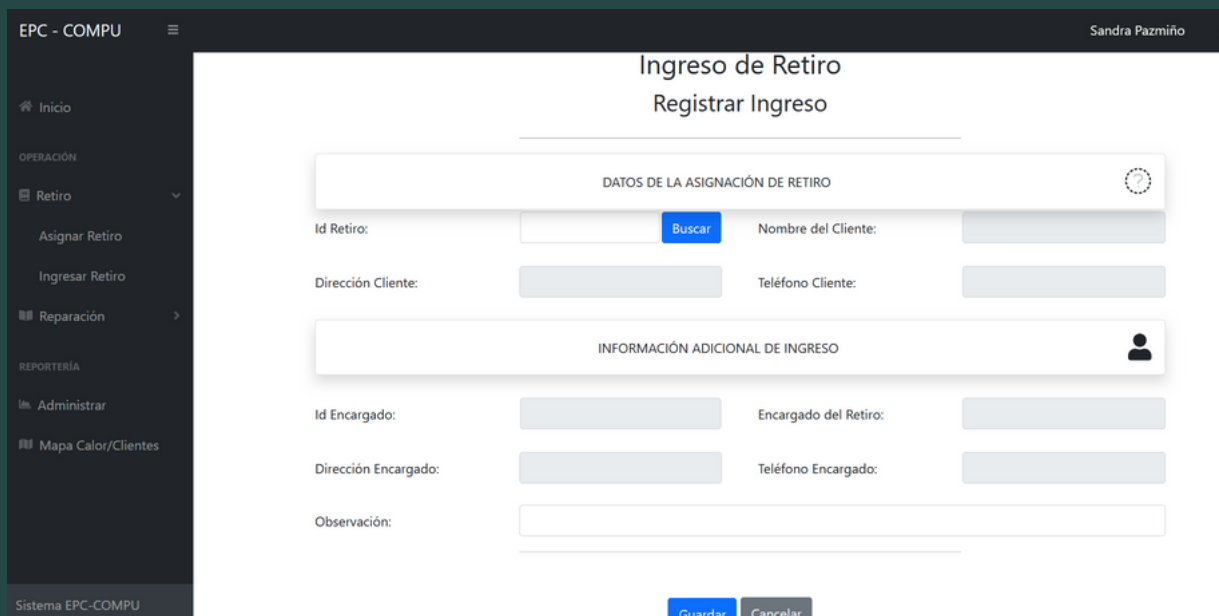
# Ingresar Retiros

Una vez seleccionado Ingresar Retiro, nos indica la siguiente pagina. En la cual dara clic en Agregar Ingreso




The screenshot shows the 'Ingreso de Retiro' page in the EPC-COMPU system. The page title is 'Ingreso de Retiro' and the user is Sandra Pazmiño. The main content area is titled 'LISTADO DE INGRESOS' and contains a search bar with the text 'Ingresos vacios'. A blue button labeled 'Agregar Ingreso' is located at the bottom right of the list area. The left sidebar menu includes 'Inicio', 'OPERACIÓN', 'Retiro' (with sub-items 'Asignar Retiro' and 'Ingresar Retiro'), 'Reparación', and 'REPORTERÍA'.

En la siguiente pagina selecciona el retiro e ingresa una observación




The screenshot shows the 'Ingreso de Retiro Registrar Ingreso' page in the EPC-COMPU system. The page title is 'Ingreso de Retiro Registrar Ingreso' and the user is Sandra Pazmiño. The form is divided into two sections: 'DATOS DE LA ASIGNACIÓN DE RETIRO' and 'INFORMACIÓN ADICIONAL DE INGRESO'. The 'DATOS DE LA ASIGNACIÓN DE RETIRO' section includes fields for 'Id Retiro', 'Nombre del Cliente', 'Dirección Cliente', and 'Teléfono Cliente', with a blue 'Buscar' button next to the 'Id Retiro' field. The 'INFORMACIÓN ADICIONAL DE INGRESO' section includes fields for 'Id Encargado', 'Encargado del Retiro', 'Dirección Encargado', 'Teléfono Encargado', and a large text area for 'Observación'. At the bottom of the form, there are 'Guardar' and 'Cancelar' buttons. The left sidebar menu is the same as in the previous screenshot, but with 'Administrar' and 'Mapa Calor/Clientes' added under 'REPORTERÍA'.

En la opción Administrar, el perfil de secretaria encuentra los siguientes informes en los cuales puede seleccionar el personal y la etapa del retiro

 Informes de Secretaria

Número de Retiros por Personal de Recepción y Entrega




**Seleccione Personal**

Fernando Naula

**Generar Informe**

Número Reparaciones por Estado

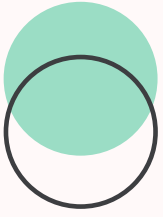


**Seleccione Estado**

Iniciado

**Generar Informe**

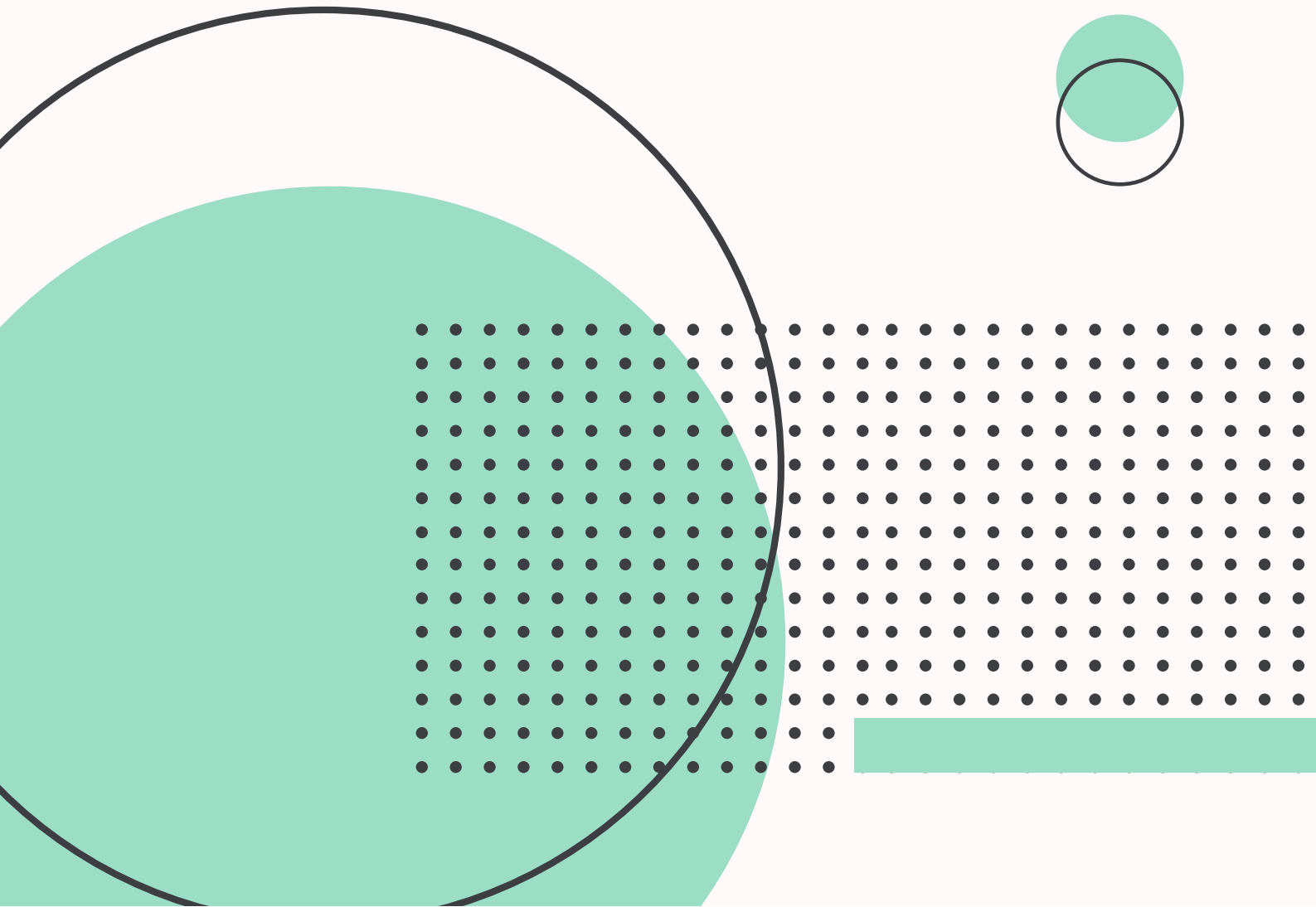
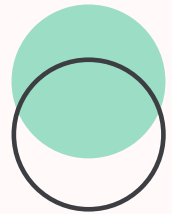
2022



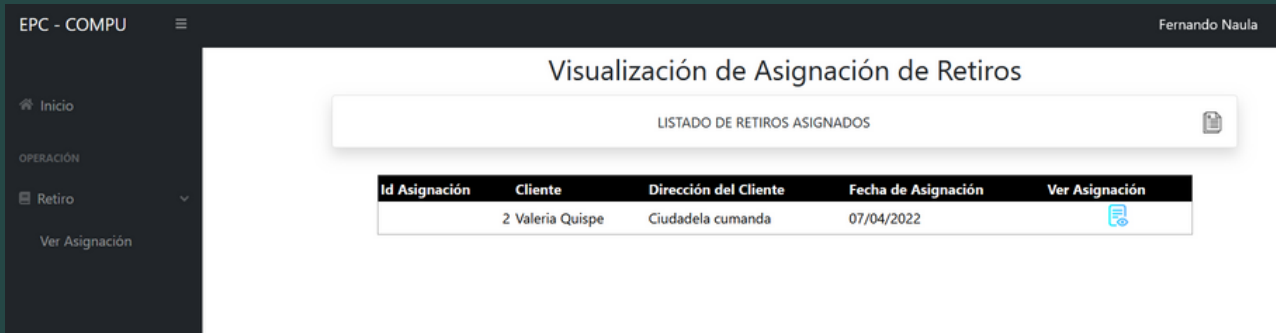
# MOTORIZADO

---

EPC COMPU



1. Ingresa las credenciales de motorizado
2. Desplegar el menu Retiro en el cual tiene la opción Ver Asignación
3. Muestra la siguiente pantalla



Presionar en el icono Ver Asignación y muestra toda la información pertinente del retiro con la respectiva ruta que debe tomar el motorizado

