



**UNIVERSIDAD CÉSAR VALLEJO**

**FACULTAD DE INGENIERÍA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**“DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C”**

TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:

Ingeniero de Sistemas

**AUTOR:**

Romero Rodríguez, Billy Jerson (0000-0001-8003-5964)

**ASESOR:**

Dr. Villaverde Medrano, Hugo (0000-0002-3802-4396)

**LÍNEA DE INVESTIGACIÓN:**

Sistemas de Información y Comunicaciones

LIMA – PERÚ

2021

## **Dedicatoria**

Este trabajo de investigación está dedicado a mi persona, ya que a pesar de las muchas dificultades que pude tener en el camino, demuestra lo mucho que uno mismo puede lograr con esfuerzo y dedicación.

## **Agradecimiento**

Agradezco a todas las personas que me apoyaron, que no dejaron que renuncie a pesar de todas las dificultades que hubo en el camino y no solo en esta etapa sino en toda mi carrera universitaria.

## Índice de contenidos

Dedicatoria.....	II
Agradecimiento .....	III
Índice de contenidos .....	IV
Índice de tablas.....	V
Índice de figuras .....	VI
Índice de anexos.....	X
Resumen.....	XI
Abstract.....	XII
I. INTRODUCCIÓN .....	1
II. MARCO TEÓRICO .....	6
III. MÉTODO.....	16
<b>3.1 Tipo y diseño de investigación</b> .....	17
<b>3.2 Variable y operacionalización</b> .....	18
<b>3.3 Población, muestra y muestreo</b> .....	19
<b>3.4 Técnicas e instrumentos de recolección de datos</b> .....	19
<b>3.5 Procedimientos</b> .....	22
<b>3.6 Método de análisis de datos</b> .....	22
<b>3.7 Aspectos éticos</b> .....	24
IV. RESULTADOS.....	25
V. DISCUSIÓN.....	33
VI. CONCLUSIONES.....	35
VII. RECOMENDACIONES .....	37
REFERENCIAS .....	39
ANEXOS .....	45

## Índice de tablas

Tabla N°1: Validación de la metodología de desarrollo de software .....	15
Tabla N°2: Instrumento para obtener los datos .....	20
Tabla N°3: Validez del instrumento de nivel de servicio .....	20
Tabla N°4: Validez del instrumento del índice de rotación de stock .....	21
Tabla N°5: Análisis descriptivo del nivel de servicio .....	26
Tabla N°6: Análisis descriptivo del índice de rotación .....	28
Tabla N°7: Prueba de Shapiro-Wilk del indicador de nivel de servicio .....	29
Tabla N°8: Prueba de Wilcoxon del indicador nivel de servicio .....	30
Tabla N°9: Estadísticos de prueba .....	30
Tabla N°10: Prueba de Shapiro-Wilk del indicador índice de rotación de stock .....	31
Tabla N°11: Prueba de Wilcoxon del indicador índice de rotación de stock .....	32
Tabla N°12: Estadísticos de prueba .....	32
Tabla N°13: Historia N°1 .....	70
Tabla N°14: Historia N°2 .....	70
Tabla N°15: Historia N°3 .....	71
Tabla N°16: Historia N°4 .....	71
Tabla N°17: Historia N°5 .....	71
Tabla N°18: Historia N°6 .....	71
Tabla N°19: Historia N°7 .....	72
Tabla N°20: Historia N°8 .....	72
Tabla N°21: Historia N°9 .....	72
Tabla N°22: Historia N°10 .....	72
Tabla N°23: Historia N°11 .....	73
Tabla N°24: Historia N°12 .....	73
Tabla N°25: Historia N°13 .....	73
Tabla N°26: Historia N°14 .....	74
Tabla N°27: Historia N°15 .....	74
Tabla N°28: Historia N°16 .....	74
Tabla N°29: Historia N°17 .....	75
Tabla N°30: Product Backlog .....	75
Tabla N°31: Product Backlog .....	76
Tabla N°32: Plan de trabajo .....	77
Tabla N°33: Sprint 1 .....	79
Tabla N°34: Cronograma del Sprint 1 .....	81
Tabla N°35: Sprint 2 .....	139
Tabla N°36: Cronograma del Sprint 2 .....	141
Tabla N°37: Sprint 3 .....	150
Tabla N°38: Cronograma del Sprint 3 .....	152
Tabla N°39: Sprint 4 .....	159
Tabla N°40: Cronograma del Sprint 4 .....	161
Tabla N°41: Sprint 5 .....	170
Tabla N°42: Cronograma del Sprint 5 .....	172

## Índice de figuras

Figura N°1. Gráfico estadístico del indicador nivel de servicio .....	4
Figura N°2. Gráfico estadístico del indicador índice de rotación de stock.....	4
Figura N°3. Diseño de estudio .....	18
Figura N°4. Interpretación de un coeficiente de confiabilidad.....	21
Figura N°5. Medida de estabilidad .....	22
Figura N°6. Indicador nivel de servicio antes y después de implementar el sistema .....	27
Figura N°7. Indicador índice de rotación de stock antes y después de implementar el sistema	28
Figura N°8. Diagrama lógico .....	82
Figura N°9. Diagrama físico .....	83
Figura N°10. Script de la tabla users .....	84
Figura N°11. Script de la tabla clientes .....	84
Figura N°12. Script de la tabla prestamos .....	84
Figura N°13. Script de la tabla proveedores .....	85
Figura N°14. Script de la tabla pedidos.....	85
Figura N°15. Script de la tabla detalle_pedidos .....	85
Figura N°16. Script de la tabla salidas .....	85
Figura N°17. Script de la tabla detalle_salidas .....	86
Figura N°18. Script de la tabla productos .....	86
Figura N°19. Script de la tabla stock_productos .....	86
Figura N°20. Script de la tabla marcas .....	86
Figura N°21. Script de la tabla estado_pedido.....	86
Figura N°22. Script de la tabla permissions .....	87
Figura N°23. Script de la tabla roles .....	87
Figura N°24. Script de la tabla model_has_permissions .....	87
Figura N°25. Script de la tabla model_has_roles.....	87
Figura N°26. Script de la tabla role_has_permissions .....	88
Figura N°27. Script de llaves foráneas de la tabla clientes y prestamos.....	88
Figura N°28. Script de llaves foráneas de la tabla pedidos y detalle_pedidos .....	88
Figura N°29. Script de llaves foráneas de la tabla salidas y detalle_salidas.....	88
Figura N°30. Script de llaves foráneas de la tabla productos y stock_productos.....	89
Figura N°31. Tabla users .....	89
Figura N°32. Tabla clientes .....	89
Figura N°33. Tabla proveedores .....	90
Figura N°34. Tabla marcas .....	90
Figura N°35. Tabla productos .....	90
Figura N°36. Tabla stock_productos .....	91
Figura N°37. Tabla pedidos.....	91
Figura N°38. Tabla detalle_pedidos .....	91
Figura N°39. Tabla salidas .....	92
Figura N°40. Tabla detalle_salidas .....	92
Figura N°41. Tabla estado_pedido .....	92
Figura N°42. Tabla prestamos .....	93
Figura N°43. Tabla roles .....	93
Figura N°44. Tabla permissions .....	93
Figura N°45. Tabla role_has_permissions .....	94
Figura N°46. Tabla model_has_roles.....	94
Figura N°47. Tabla model_has_permissions .....	94
Figura N°48. Prototipo Login .....	95
Figura N°49. Maquetación de la interfaz del Login .....	95
Figura N°50. Prototipo principal del sistema .....	96
Figura N°51. Maquetación de la interfaz principal del sistema .....	96
Figura N°52. Prototipo del módulo usuarios .....	97
Figura N°53. Maquetación del módulo usuarios .....	97
Figura N°54. Prototipo del módulo usuarios - registrar .....	98

<i>Figura N°55. Maquetación del módulo usuarios - registrar</i> .....	98
<i>Figura N°56. Prototipo del módulo usuarios - editar</i> .....	99
<i>Figura N°57. Maquetación del módulo usuarios - editar</i> .....	99
<i>Figura N°58. Prototipo del módulo usuarios - eliminar</i> .....	100
<i>Figura N°59. Maquetación del módulo usuarios - eliminar</i> .....	100
<i>Figura N°60. Prototipo del módulo cliente</i> .....	101
<i>Figura N°61. Maquetación del módulo cliente</i> .....	101
<i>Figura N°62. Prototipo del módulo cliente - registrar</i> .....	102
<i>Figura N°63. Maquetación del módulo cliente - registrar</i> .....	102
<i>Figura N°64. Prototipo del módulo cliente - editar</i> .....	103
<i>Figura N°65. Maquetación del módulo cliente - editar</i> .....	103
<i>Figura N°66. Prototipo del módulo cliente - eliminar</i> .....	104
<i>Figura N°67. Maquetación del módulo cliente - eliminar</i> .....	104
<i>Figura N°68. Prototipo del módulo proveedores</i> .....	105
<i>Figura N°69. Maquetación del módulo proveedores</i> .....	105
<i>Figura N°70. Prototipo del módulo proveedores - registrar</i> .....	106
<i>Figura N°71. Maquetación del módulo proveedores - registrar</i> .....	106
<i>Figura N°72. Prototipo del módulo proveedores - editar</i> .....	107
<i>Figura N°73. Maquetación del módulo proveedores - editar</i> .....	107
<i>Figura N°74. Prototipo del módulo proveedores - eliminar</i> .....	108
<i>Figura N°75. Maquetación del módulo proveedores - eliminar</i> .....	108
<i>Figura N°76. Prototipo del módulo marcas</i> .....	109
<i>Figura N°77. Maquetación del módulo marcas</i> .....	109
<i>Figura N°78. Prototipo del módulo marcas - registrar</i> .....	110
<i>Figura N°79. Maquetación del módulo marcas - registrar</i> .....	110
<i>Figura N°80. Prototipo del módulo marcas - editar</i> .....	111
<i>Figura N°81. Maquetación del módulo marcas - editar</i> .....	111
<i>Figura N°82. Prototipo del módulo marcas - eliminar</i> .....	112
<i>Figura N°83. Maquetación del módulo marcas - eliminar</i> .....	112
<i>Figura N°84. Prototipo del módulo productos</i> .....	113
<i>Figura N°85. Maquetación del módulo productos</i> .....	113
<i>Figura N°86. Prototipo del módulo productos - registrar</i> .....	114
<i>Figura N°87. Maquetación del módulo productos - registrar</i> .....	114
<i>Figura N°88. Prototipo del módulo productos - editar</i> .....	115
<i>Figura N°89. Maquetación del módulo productos - editar</i> .....	115
<i>Figura N°90. Prototipo del módulo productos - eliminar</i> .....	116
<i>Figura N°91. Maquetación del módulo productos - eliminar</i> .....	116
<i>Figura N°92. Prototipo del módulo entrada de productos</i> .....	117
<i>Figura N°93. Maquetación del módulo entrada de productos</i> .....	117
<i>Figura N°94. Prototipo del módulo entrada de productos - registrar</i> .....	118
<i>Figura N°95. Maquetación del módulo entrada de productos - registrar</i> .....	118
<i>Figura N°96. Prototipo del módulo entrada de productos – lista de productos</i> .....	119
<i>Figura N°97. Maquetación del módulo entrada de productos – lista de productos</i> .....	119
<i>Figura N°98. Prototipo del módulo entrada de productos – agregar producto</i> .....	120
<i>Figura N°99. Maquetación del módulo entrada de productos – agregar producto</i> .....	120
<i>Figura N°100. Prototipo del módulo entrada de productos – editar</i> .....	121
<i>Figura N°101. Maquetación del módulo entrada de productos – editar</i> .....	121
<i>Figura N°102. Prototipo del módulo entrada de productos – eliminar</i> .....	122
<i>Figura N°103. Maquetación del módulo entrada de productos – eliminar</i> .....	122
<i>Figura N°104. Prototipo del módulo salida de productos</i> .....	123
<i>Figura N°105. Maquetación del módulo salida de productos</i> .....	123
<i>Figura N°106. Prototipo del módulo salida de productos - registrar</i> .....	124
<i>Figura N°107. Maquetación del módulo salida de productos - registrar</i> .....	124
<i>Figura N°108. Prototipo del módulo salida de productos – lista de productos</i> .....	125
<i>Figura N°109. Maquetación del módulo salida de productos – lista de productos</i> .....	125
<i>Figura N°110. Prototipo del módulo salida de productos – agregar producto</i> .....	126

Figura N°111. Maquetación del módulo salida de productos – agregar producto.....	126
Figura N°112. Prototipo del módulo salida de productos – editar.....	127
Figura N°113. Maquetación del módulo salida de productos – editar.....	127
Figura N°114. Prototipo del módulo salida de productos – eliminar.....	128
Figura N°115. Maquetación del módulo salida de productos – eliminar.....	128
Figura N°116. Prototipo del módulo préstamos.....	129
Figura N°117. Maquetación del módulo préstamos.....	129
Figura N°118. Prototipo del módulo préstamos - registrar.....	130
Figura N°119. Maquetación del módulo préstamos - registrar.....	130
Figura N°120. Prototipo del módulo préstamos - editar.....	131
Figura N°121. Maquetación del módulo préstamos - editar.....	131
Figura N°122. Prototipo del módulo préstamos - eliminar.....	132
Figura N°123. Maquetación del módulo préstamos - eliminar.....	132
Figura N°124. Prototipo del módulo reporte.....	133
Figura N°125. Maquetación del módulo reporte.....	133
Figura N°126. Prototipo del módulo reporte – generar reporte.....	134
Figura N°127. Maquetación del módulo reporte – generar reporte.....	134
Figura N°128. Código para validar las columnas de la tabla usuarios.....	135
Figura N°129. Código para listar los registros la tabla usuarios.....	135
Figura N°130. Código para realizar un registro en la tabla usuarios.....	136
Figura N°131. Código para editar un registro en la tabla usuarios.....	136
Figura N°132. Código para eliminar un registro en la tabla usuarios.....	137
Figura N°133. Integración del código a la interfaz del login.....	137
Figura N°134. Creación de permisos para cada rol.....	142
Figura N°135. Permisos predefinidos - administrador.....	142
Figura N°136. Permisos predefinidos - usuario.....	143
Figura N°137. Cliente - Registrar.....	143
Figura N°138. Cliente - Editar.....	144
Figura N°139. Cliente - Eliminar.....	144
Figura N°140. Cliente - Interfaz.....	144
Figura N°141. Cliente – Ingreso de datos.....	145
Figura N°142. Proveedor - Registrar.....	145
Figura N°143. Proveedor - Editar.....	146
Figura N°144. Proveedor - Eliminar.....	146
Figura N°145. Proveedor - Interfaz.....	146
Figura N°146. Proveedor – Ingreso de datos.....	147
Figura N°147. Marca - Registrar.....	147
Figura N°148. Marca - Editar.....	147
Figura N°149. Marca - Eliminar.....	148
Figura N°150. Marca - Interfaz.....	148
Figura N°151. Marca – Ingreso de datos.....	148
Figura N°152. Producto - Registrar.....	153
Figura N°153. Producto - Editar.....	153
Figura N°154. Producto - Eliminar.....	153
Figura N°155. Producto - Interfaz.....	154
Figura N°156. Producto – Ingreso de datos.....	154
Figura N°157. Alerta de stock - Script.....	155
Figura N°158. Alerta de stock – Diseño.....	155
Figura N°159. Alerta de stock – Muestra de datos.....	155
Figura N°160. Entrada de producto - Registrar.....	156
Figura N°161. Entrada de producto - Editar.....	156
Figura N°162. Entrada de producto - Eliminar.....	157
Figura N°163. Entrada de producto - Interfaz.....	157
Figura N°164. Entrada de producto – Ingreso de datos.....	157
Figura N°165. Salida de producto - Registrar.....	162
Figura N°166. Salida de producto - Editar.....	162



<i>Figura N°167. Salida de producto - Eliminar</i> .....	162
<i>Figura N°168. Salida de producto - Interfaz</i> .....	163
<i>Figura N°169. Salida de producto – Ingreso de datos</i> .....	163
<i>Figura N°170. Lista de productos – Registrar</i> .....	164
<i>Figura N°171. Lista de productos – Editar</i> .....	164
<i>Figura N°172. Lista de productos – Eliminar</i> .....	165
<i>Figura N°173. Lista de productos – Interfaz</i> .....	165
<i>Figura N°174. Lista de productos – Ingreso de datos</i> .....	166
<i>Figura N°175. Prestamos - Registrar</i> .....	166
<i>Figura N°176. Prestamos - Editar</i> .....	167
<i>Figura N°177. Prestamos - Eliminar</i> .....	167
<i>Figura N°178. Prestamos - Interfaz</i> .....	167
<i>Figura N°179. Prestamos – Ingreso de datos</i> .....	168
<i>Figura N°180. Reportes – Procesar fecha</i> .....	173
<i>Figura N°181. Reportes – Ingreso de la fecha</i> .....	173
<i>Figura N°182. Reportes – Interfaz</i> .....	174
<i>Figura N°183. Confirmación de cambio de datos</i> .....	174
<i>Figura N°184. Mensaje de confirmación</i> .....	174
<i>Figura N°185. Validación de datos</i> .....	175
<i>Figura N°186. Validación de datos - Interfaz</i> .....	175
<i>Figura N°187. Comprobación del stock con la BD</i> .....	176
<i>Figura N°188. Validación de stock - Script</i> .....	176

## Índice de anexos

<b>Anexo N°1: Matriz de operacionalización de variables</b> .....	46
<b>Anexo N°2: Matriz de consistencia</b> .....	47
<b>Anexo N°3: Instrumento de recolección de datos</b> .....	48
<b>Anexo N°4: Calculo del tamaño de la muestra</b> .....	50
<b>Anexo N°5: Validación del instrumento de investigación</b> .....	51
<b>Anexo N°6: Resultados de la confiabilidad de los instrumentos</b> .....	57
<b>Anexo N°7: Constancia de autorización para realizar el trabajo de investigación</b> .....	61
<b>Anexo N°8: Entrevista al gerente de la empresa Selva Gas Tocache S.A.C</b> .....	62
<b>Anexo N°9: Proceso de control de inventario</b> .....	64
<b>Anexo N°10: Validación de la metodología</b> .....	65
<b>Anexo N°11: Constancia de aprobación de la empresa</b> .....	68
<b>Anexo N°12: Desarrollo de la metodología de Scrum</b> .....	69

## Resumen

El siguiente trabajo de investigación contiene el desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C. El tipo de investigación que se utilizó fue la aplicada mediante el estudio de datos cuantitativos, así mismo se hizo uso del diseño de investigación experimental.

El objetivo general fue determinar la influencia del sistema web en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C. Mediante la validación de expertos, se determinó que la metodología a usar fue SCRUM; además se utilizó el lenguaje de programación PHP, para la base de datos se usó MYSQL, también se usó el framework Laravel y entre otras cosas.

Referente al indicador nivel de servicio la población fue de 540 pedidos y su muestra fue de 30 días, por parte del indicador índice de rotación de stock la población fue de 600 productos y su muestra fue de 30 días. Para la prueba de normalidad se utilizó el método de Shapiro – Wilk ya que la muestra es menor a 50; debido a que los datos obtenidos no se distribuyen de forma normal se utilizó la prueba de rangos de Wilcoxon para determinar si se acepta o rechaza la hipótesis.

Los resultados obtenidos para el indicador nivel de servicio en el pre – test fue de 76.23% y el post – test fue de 90.83%, y para el indicador índice de rotación de stock en el pre – test fue de 23.7% y el post – test fue de 42.5%. Con respecto a los datos mencionados anteriormente se puede concluir que el sistema web mejoró el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

Palabras claves: Sistema web, proceso de control de inventario, nivel de servicio, índice de rotación de stock, scrum.

## **Abstract**

The following research work contains the development of a web system for the inventory control process in the company Selva Gas Tocache S.A.C. The type of research that was used was applied through the study of quantitative data, likewise the experimental research design was used.

The general objective was to determine the influence of the web system in the inventory control process in the company Selva Gas Tocache S.A.C. Through the validation of experts, it was determined that the methodology to be used was SCRUM; In addition, the PHP programming language was used, MYSQL was used for the database, the Laravel framework was also used, and among other things.

Regarding the service level indicator, the population was 540 orders and its sample was 30 days, on the part of the stock rotation index indicator the population was 600 products and its sample was 30 days. For the normality test, the Shapiro-Wilk method was used since the sample is less than 50; Since the data obtained are not normally distributed, the Wilcoxon rank test was used to determine whether the hypothesis is accepted or rejected.

The results obtained for the service level indicator in the pre - test was 76.23% and the post - test was 90.83%, and for the stock turnover index indicator in the pre - test it was 23.7% and the post - test was 42.5%. Regarding the aforementioned data, it can be concluded that the web system improved the inventory control process in the company Selva Gas Tocache S.A.C.

Keywords: Web system, inventory control process, service level, stock turnover rate, scrum.

# **I. INTRODUCCIÓN**

A nivel internacional, según Rodríguez, Salazar y Gonzáles (2018) citado por Samaniego (2019), nos menciona que la gestión de inventarios en muchos casos está limitada por las habilidades y conocimientos de los trabajadores encargados de este asunto, como por ejemplo son los administradores, y que a menudo no logran conseguir los objetivos establecidos por la organización que en grandes rasgos es tener un stock equilibrado lo cual es un asunto complejo para las empresas.

A nivel nacional, Vidal Holguin, (2020), nos menciona que en cuanto a los problemas sobre el control de inventarios se pueden dividir dependiendo de las características de una demanda y sobre los tiempos de reposición. Estos 2 se pueden clasificar en determinísticos o aleatorios, y entre estos la demanda se puede dividir en constante y conocida, demanda determinística y demanda probabilística o aleatoria. Con respecto a la primera demanda su estudio facilita el entendimiento de procesos más complejos, en cuanto a la segunda demanda su conocimiento es más preciso ya que se puede llegar a conocer antes de que ocurra, un ejemplo serían los contratos de venta preestablecidos, y por último concerniente a la tercera demanda se asume parcialmente que los tiempos de reabastecimientos serán constantes y conocidos.

Según Anzola (2010) citado por Carrillo, Bravo, y Córdova (2016), menciona que la falta de control de inventarios no es un problema que afecta solo a las grandes empresas, también es un de los principales problemas por lo que atraviesan las pequeñas empresas, esto se debe tratar con prudencia y dependiendo de la circunstancia ya que ponen en riesgo el no cumplir a tiempo con sus compromisos que tienen con sus clientes.

El presente trabajo de investigación se realizará en la empresa Selva Gas Tocache S.A.C está ubicado en el departamento de San Martín y tiene como gerente general a la Ing. Estupiñan Palma, Patricia Lesly, dueña de la empresa la cual brinda servicio de venta de balones de gas. Asimismo, la empresa en los últimos años su inventario está en auge por lo cual ha estado realizando un abastecimiento de sus productos semanalmente para una clientela de 80 a 100 clientes mensuales aproximadamente. Esto ha causado malestar a la gerencia y a los trabajadores con respecto al inventario de los productos, concordando con

Mendoza (2019), indicando que no llevar un buen control del inventario conlleva a tener una baja rentabilidad y así mismo a tener un impacto negativo en las ventas.

Por otro lado, las ventas de balones de gas están en auge ya que en la ciudad de Tocache no hay un abastecimiento de gas total al 100% ni llegando a superar el 40%, según Cantuarias (2020), vocero del gremio de SPGL manifiesta que en el año 2011 sólo 4 regiones tenían locales de venta de GLP que cubrían el 40% de sus distritos, pero el año 2020 hubo un aumento exponencial de tener sólo 4 regiones a tener 21 regiones con locales de venta de GLP y que cubren más del 40% de sus distritos. Siendo esto una oportunidad de ampliar el nicho de mercado para la empresa.

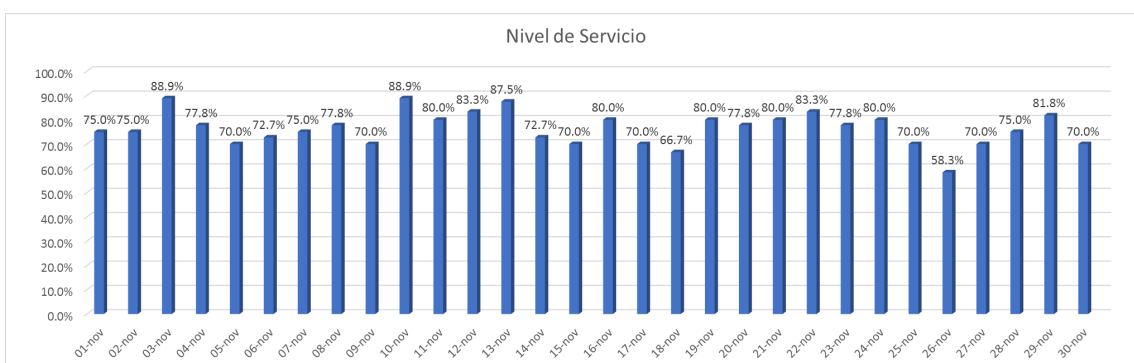
Debido a esto, la empresa ha presentado inconvenientes con los pedidos al vender a través de las redes sociales, al no contar con stock a afectado su reputación, por tanto, la deficiente gestión de inventario a afectado en la atención de los pedidos coincidiendo con Follegati (2019) nos menciona que hay otras formas de ofertar los productos que una empresa o negocio oferta, y una de esas es a través de plataformas online, pero esta es una herramienta de doble filo, ya que si no se tiene una gestión adecuada del inventario, ocasionará molestias a sus clientes como por ejemplo recibir pedidos de un producto que ya se agotó y afectará negativamente a su reputación.

En una entrevista con la gerente general (Anexo 10), nos menciona que el proceso de venta que realizan es simplemente, atender al cliente cuando solicita un producto, el trabajador no tiene como verificar en ese momento si tiene stock del producto que desea el cliente y por este motivo tiene que ir al almacén y verificar si tienen stock, luego se procede a realizar la venta; cuando se genera la venta no llevan el registro sobre qué productos salen y la cantidad de éste. También realizan préstamos los cuales llevan un registro físico (apuntes en cuadernos), que no es muy seguro puesto que con el tiempo dichos registros se pueden deteriorar o extraviar. Esta falta de control del inventario ocasiona problemas con los clientes, ya que al no saber con exactitud el stock de los productos se llevan a perder ventas y lo más importante, la imagen que dan hacia

los clientes, la cual es ser una empresa sin organización y que brinda un pésimo servicio.

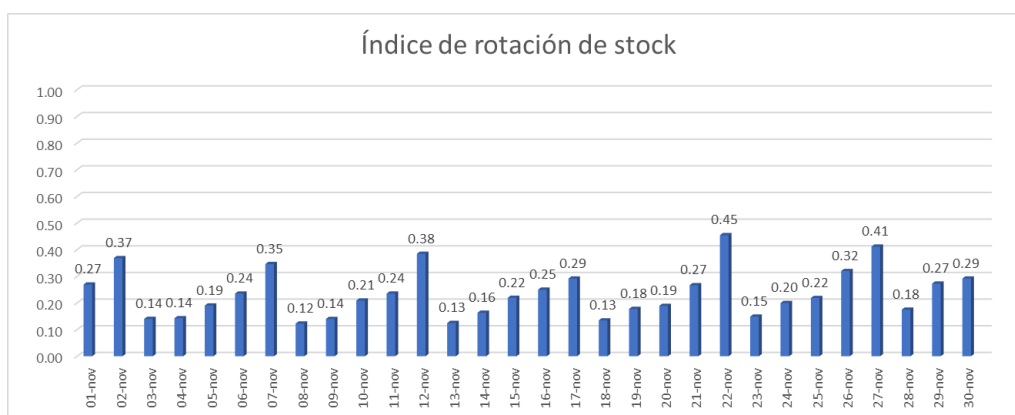
Como se muestra a continuación en la en la figura N°1 del indicador nivel de servicio, los datos proporcionados por la empresa entre el 1 noviembre hasta el 30 de noviembre que serán utilizados para realizar los procesos estadísticos correspondientes como se muestra en el Anexo N°5.

Figura N°1. Gráfico estadístico del indicador nivel de servicio.



De igual manera para en la figura N°2 se muestran los datos proporcionados por la empresa para el indicador índice de rotación de stock entre el 1 de noviembre hasta el 30 de noviembre que serán utilizados para realizar los procesos estadísticos correspondientes como se muestra en el Anexo N°5.

Figura N°2. Gráfico estadístico del indicador índice de rotación de stock.



Por lo tanto, se propuso resolver esta problemática mediante el diseño, desarrollo e implementación de un sistema web que mejorara el proceso del control de inventario en la empresa Selva Gas Tocache S.A.C y de esta manera puedan estar más organizados al momento de brindarle el servicio al cliente.



En base a ello, el objetivo general de la esta investigación es: Determinar la influencia del sistema web en el proceso de control de inventario de la empresa Selva Gas Tocache S.A.C. Asimismo, el primer objetivo específico fue: Determinar la influencia del sistema web en el nivel de servicio para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C y el segundo objetivo fue: Determinar la influencia del sistema web en el índice de rotación de stock para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

Por último, la hipótesis general de la investigación fue: El sistema web mejora el proceso de control de inventario de la empresa Selva Gas Tocache S.A.C. Sumado a ello, la primera hipótesis específica es: El sistema web mejora el nivel de servicio en el proceso de control de inventario de la empresa de Seva Gas Tocache S.A.C y la segunda hipótesis específica es: El sistema web mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

## **II. MARCO TEÓRICO**

La presente investigación tiene en el ámbito nacional los siguientes trabajos previos:

Hernandez Castañeda (2019), menciona en su investigación que la finalidad de la investigación fue implementar un sistema web de control de inventarios que permita mejorar el sistema actual, ya que éste es muy antiguo, y no logra satisfacer las necesidades de la institución. Se tomo como muestra 12 personas, el tipo de investigación que realizó fue la aplicada con un nivel de tipo descriptivo que a su vez esta compuesto por un diseño pre-experimental. Se utilizó la metodología RUP en el proceso de desarrollo del sistema y de esta manera lograr cumplir los objetivos que se plantearon. Se utilizó la arquitectura MVC (Modelo Vista Controlador) a travez del IDE RubyMune, se uso el lenguaje de programación Ruby, el motor de base de datos PostgreSQL, Bootstrap para los estilos de diseño o responsivo y usando el framework Web Rails. La tesis nos presenta como resultados que se consiguió comprobar que al finalizar e implementar el nuevo sistema web influyó positivamente en el control y manejo de los bienes patrimoniales de la institución, manifestando estadísticamente una gran mejoría en el nivel de servicio hacia los usuarios, mejora en la seguridad de los datos, reducción de tiempo al momento de realizar un registro o reporte y una mejoría en la gestión de inventarios.

Monteza Torres (2019), menciona en su investigación que su finalidad fue realizar un sistema web de gestión de almacén para mejorar dicho proceso y generar una mayor productividad a corto plazo en la empresa. La población y muestra fue de 35 trabajadores, tuvo un tipo de investigación descriptiva – explicativo y de diseño pre – experimental. Se establecio que la metodología mas adecuada para el desarrollo del sistema es RUP, para la programación del código se uso el lenguaje PHP, Java y para el diseño de los estilos el lenguaje CSS, para la base de datos se implemento SQL Server. La investigación muestra, luego de procesar los datos arrojados por las encuestas, que al utilizar el sistema web brindó una mejor accesibilidad de la información proporcionada por los procesos de control de entrada y salidas, por otro lado se contemplo un incremento mayor al 90% en la gestión del almacén.

Según Purisaca Martínez y Zavaleta Velásquez (2019), nos mencionan que su investigación tuvo como finalidad comprobar que utilizando tecnologías de desarrollo y metodologías ágiles se consigue una mejoría en los procesos de control de materiales arqueológicos en el transcurso de proyectos de excavación, y para ello recurrieron a tarjetas CRC y el uso de la metodología SCRUM. La tesis nos presenta como resultado que al realizar la sistematización en el control de inventario se logró un control más automatizado y de esta manera realizar de forma más eficiente la entrada de materiales, generar consultas y elaborar reportes de forma más fácil.

Beltran Castillo (2017), nos menciona en su investigación que la creación y uso de un sistema web logra generar un mayor control de los inventarios y de la asignación que tuvieron los bienes. Su población y muestra fue de 18 empleados, la investigación es de tipo descriptiva y las medidas realizadas son cuantitativas, el diseño de la investigación fue no experimental, la metodología híbrida que utilizaron para desarrollar el software fue ICONIX. La tesis nos presenta como resultado que se obtuvo un 100% de las encuestas realizado a los empleados, lo cual representa la urgencia de crear un sistema web con influencia positiva en el área de control de inventarios, puesto que los empleados no están satisfechos de cómo se está realizando el trabajo actualmente en la empresa.

La presente investigación tiene en el ámbito internacional los siguientes trabajos previos:

Lucas Vega (2017), menciona en su investigación el desarrollar una aplicación web para disminuir la gran carga operativa que se realiza en el inventario de mercaderías de las diferentes sucursales. El sistema se realizó con el lenguaje node.js, una base de datos MySQL 5 con una arquitectura MVC, el cual permitirá una búsqueda de información más fluida. La tesis nos da como resultado que la implementación del sistema web resolvió la dificultad por la que pasaban los usuarios, así mismo da la oportunidad de integrar nuevos módulos a la herramienta tecnológica y de esta manera tener la oportunidad de crecer mucho más.

Lozano Briones (2017), describe en su investigación el desarrollar una aplicación web que ayude a la automatización del proceso Kardex, y de esta forma mejore la gestión que se realiza en los almacenes de la compañía. Para ello se utilizó la metodología ágil Scrum y para la parte del Front-end se dio uso a las herramientas que tienen una distribución libre tales como el lenguaje de maquetación HTML5, CSS3, Bootstrap, Ajax, JQuery y para la parte del Back-end se uso el framework Symfony, como base de datos se uso el MySQL con servidor web apache y otras cosas más. Entre los módulos más importantes observamos la gestión de registro de entrada y salida de los productos, control del inventario, características de los productos, consulta de los datos de los diferentes productos, solicitud de algún producto que sera enviado hacia alguna localidad y también se podra consultar los diferentes movimientos que se hallan realizado de los diferentes productos. En conclusión el resultado es un producto de software eficaz del cual se pudo comprobar la eficacia de la metodolofía aplicada, además la implementación a la compañía fue sencilla y otro aspecto importante es su agilidad que demuestra a los cambios, y con ello se cumplen los objetivos que se manifestaron al inicio de la investigación.

Vera Yáñez (2019), menciona como se realizó un sistema que provea gestionar de forma óptima el control del proceso de alquiler de la mercancía e inventarios. Para esta investigación el diseño de investigación fue aplicada y de tipo experimental. En el desarrollo del sistema se dio uso al lenguaje PHP 7 en el backend, el motor de base de datos que se usó fue MariaDB y en el frontend se utilizó JQuery, Bootstrap y entre otras cosas. La tesis como conclusion nos menciona que el sistema web desarrollado permitio centralizar la información y con ello generar una mayor productividad en el proceso de alquiler de mercancías e inventarios.

Martin Romero (2019), nos menciona que su investigación tuvo como finalidad el desarrollo de una aplicación web que proporcione un mayor control de los diferentes productos que la empresa comercializa, puesto que la gestión actual en estos proceso se realizan a través de hojas de calculo y anotaciones, lo cual es muy inadecuada e ineficiente. Para el desarrollo del sistema web se hizo uso de la metodología ágil SCRUM, para la base de datos se uso MySQL, se utilizó el

estandar JAVA EE y como framework el JSF. La tesis nos menciona en su conclusión que todas las herramientas que se usaron generó una aplicación de gestión de inventario eficaz el cual proporcionara de manera más clara la información sobre el estado de stock de los productos en el inventario.

En adición, la presente investigación contiene las siguientes teorías relacionadas:

La variable independiente, sistema web, también llamadas aplicaciones web, su desarrollo va centrada a cumplir las necesidades de automatización de las organizaciones que cuentan con una considerable generación de datos. Los sistemas web son alojados en un hosting, para que así se pueda ingresar desde cualquier sitio que cuenta con conexión a internet, aparte también puede estar alojado en el localhost. De igual manera, los siguientes autores nos mencionan lo siguiente:

Una aplicación web es un programa de software de tipo cliente-servidor que se divide en dos partes, la primera es el back-end la cual se ejecuta en un servidor web, mientras que la otra parte es decir el front-end, que es la parte visual en la cual el usuario realiza las distintas acciones que proporciona dicha aplicación web, se ejecutan a través de un navegador web (Lotfy y Pyatt, 2018, p. 99).

Se conoce como aplicación web al software que reside en una computadora, que también es llamado servidor web, el cual es utilizado por los usuarios a través de internet haciendo uso de algún navegador web, y así poder obtener los distintos servicios que pueda ofrecer dicha aplicación web (Jiménez, 2013, p. 8).

Hay una variedad de aplicaciones web y que pueden ser de diferentes tipos, tales como administradores de correos, tiendas online, blogs, etc. A su vez estas aplicaciones web tienen dos tipos de accesos que pueden ser públicas y restringidas. Referente a las públicas, en su totalidad es la mayoría de las aplicaciones web, como los diarios digitales, tiendas virtuales, etc. Con respecto a las restringidas están las intranets que se encargan de mejorar las gestiones internas de alguna empresa en específico, como administrar la asistencia de sus empleados, gestionar los documentos, etc. Entre las restringidas también están

las extranets cuyo objetivo es mejorar el servicio con sus clientes, colaboradores externos, distribuidores o proveedores (Jiménez, 2013, p. 12).

Aparte del sistema web, para desarrollarlo se necesita aplicar un tipo de lenguaje de programación necesario. Los siguientes autores opinan lo siguiente:

El lenguaje de programación PHP es de código abierto y que en los últimos años está siendo muy popular, puesto que es adecuado para el desarrollo de sistemas en el entorno web y que a su vez encaja con el uso del HTML (PHP GROUP, 2021).

Una de sus características más importantes que nos brinda el uso del lenguaje de programación PHP es el soporte que brinda a distintas bases de datos, además de su extrema facilidad hacia los principiantes, pero para aquellos programadores profesionales también les ofrece distintas características mucho más avanzadas (PHP GROUP, 2021).

Java es uno de los lenguajes de programación más usados en el ámbito empresarial, puesto que sus aplicaciones de gestión tienen buen nivel de escalabilidad y disponibilidad. Además, dispone de conocimientos hacia la programación orientada a objetos, como también hacia la arquitectura de software (Ortega, 2018).

El lenguaje de programación Python fue creado por Guido Van Rosum en los 90's y señala que su nombre proviene del comic Monty Python. Este lenguaje nos proporciona una sintaxis legible y limpia, además de poseer un tipado dinámico, es decir que sus variables pueden tener diferentes tipos de datos, y una de sus grandes ventajas es que su código no necesita ser compilado para ser ejecutado (Nolasco, 2018, p. 14).

Además, es necesario un gestor de base de datos donde se almacenará la información que el sistema procesará. A continuación, se mencionan algunos:

La base de datos más utilizada y conocida por los programadores de PHP es el MySQL, soporta tanto la conectividad de una variedad de usuarios como el lenguaje SQL, además se utiliza mayormente en la elaboración de aplicaciones web de tamaño pequeño-mediano (Pavón y Llarena, 2015, p. 17).

El gestor de base de datos relacionales PostgreSQL, es un potente sistema de código abierto, que cuenta con más de 30 años de desarrollo eficaz y por ello se ha ganado la reputación de ser confiable y tener una gran solidez en funciones y rendimiento (Grupo de desarrollo global de PostgreSQL, 2021).

El sistema PostgreSQL es compatible con una gran variedad del estándar SQL, el cual posee una diversidad de características modernas: integridad transaccional, control de concurrencia multiversion, llaves extranjeras, consultas complejas. Así mismo el usuario puede ampliar PostgreSQL, por ejemplo, agregando operadores, funciones, lenguajes procedimentales, métodos de índice. Y debido a su licencia liberal, puede ser usado para cualquier propósito ya sea académico, empresarial o privado (Grupo de desarrollo global de PostgreSQL, 2021).

La base de datos SQL Server nos proporciona una gran variedad de tablas en donde se pueden almacenar datos estructurados. Dicha tabla contiene una colección de tuplas o también llamado registros y columnas o también conocidos como atributos, cada columna de una tabla está diseñado para almacenar tipo específico de dato, ya sea una fecha, número o nombres (Microsoft, 2017)

Por otro lado, el framework permite realizar el desarrollo del sistema de manera más organizada y eficiente. A continuación, los autores dan a entender que:

Laravel es un contorno de la aplicación web con una interacción expresiva y elegante. Esto proporciona una distribución y una línea de partida para el desarrollo de la aplicación, lo que favorece a la fijación de los detalles (Otwell, 2011).

Laravel es muy escalable, tanto así que pueden fácilmente realizar el manejo de cientos de millones de solicitudes por mes, y esto se debe a su naturaleza amigable con el escalado de PHP y al soporte integrado de Laravel para los diferentes sistemas de caché distribuidos y rápidos (Otwell, 2011).

Por otro lado, la variable dependiente, proceso de control de inventario, es un proceso que se encarga de controlar el stock de los productos de una



organización para que de esta manera puedan satisfacer las necesidades del cliente. Asimismo, los autores nos mencionan que:

Para poder gestionar de manera adecuada los inventarios se debe dar seguimiento a los siguientes pasos: identificar aquellos productos que serán registrados y posteriormente la organización controlara en los inventarios, determinar cuáles serán los productos que se registrarán en el inventario y controlar los lugares en los que estarán presentes, reunir un grupo de empleados que se encargarán de gestionar los almacenes y por último realizar su respectivo análisis y registro estableciendo una fecha específica para su correcta realización (Fresneda, 2019).

La gestión de inventarios tiene un gran impacto en las empresas ya que indican directamente en el nivel de servicio y también en el tiempo de respuesta hacia los clientes, y esto se refleja en los ingresos. En relación a los costos, se puede materializar una buena gestión de inventarios en disminución de costos sin perjudicar el nivel de servicio lo cual es clave actualmente para las empresas (Cardon, 2018, p. 194).

Las fases del proceso de control de inventario son las siguientes:

Fase 1: Compras, esta fase facilita los componentes indispensables para la producción, los cuales son adquiridos de los proveedores sean materias primas o productos finales, en la cantidad requerida y al menor costo posible.

Fase 2: Recepción, se ocupa de las entradas de mercancías, descarga y verificación, para que los registros de inventario estén actualizados.

Fase 3: Almacén, una vez recepcionada la mercancía se procede a su ubicación y conservación (teniendo en cuenta si se trata de materia prima o productos finales) con el fin de realizar las operaciones de la empresa con éxito.

Fase 4: Entrega, en esta fase se despacha los productos requeridos por los clientes, tomando en cuenta que se debe llevar una anotación de las salidas de los artículos de almacén y así contar con un inventario actualizado (Rincón y Villarreal, 2014, p. 65).

Asimismo, la presente investigación tiene los siguientes indicadores, según su dimensión:

Dimensión almacén: indicador índice de rotación de stock. Esto hace referencia al número de veces que un producto o artículo pasa por el proceso de ser vendido, salir del almacén y ser cobrado, en un determinado periodo de tiempo y así recuperar la inversión que se realizó al adquirirlo (Aymerich, 2020, p. 58).

Cuando realizamos el cálculo del número de veces que se ha vendido un producto o artículo en un periodo de tiempo, se puede llegar a calcular la rotación de dichos productos (Aymerich, 2020, p. 60).

$$RPT = \frac{\textit{Ventas a precio de coste}}{\textit{Saldo medio de productos terminados}}$$

Dimensión entrega: indicador nivel de servicio. Lograr tener un nivel de servicio alto conlleva tener mayores costos concerniente a la gestión de almacenamiento y pedidos, esta decisión está en función a qué imagen se pretende tener (Arenal, 2020).

La gestión del inventario en una empresa influye en el nivel de servicio que pretenda dar, si a causa de una falla en el stock, una parte de la demanda queda insatisfecha, es razonable pensar que esos clientes busquen ese mismo producto o artículo en la competencia. Al referirnos al nivel de servicio estamos hablando sobre la satisfacción que proporciona una empresa hacia sus clientes. Contar con un buen nivel de servicio, significa que cualquier artículo o producto buscado por un cliente, lo encuentre en ese momento (Arenal, 2020).

$$\textit{Nivel de servicio} = \frac{\textit{Ventas}}{\textit{Demanda}} * 100$$

Por otro lado, los proyectos para desarrollar un software utilizan una metodología de desarrollo para planificar, controlar, ejecutar y concretar la finalización del sistema de forma adecuada. A continuación, los autores mencionan algunas:

La metodología RUP está centrado en el desarrollo de proyecto de aplicaciones y software de larga duración. Así mismo, es uno de principales marcos de trabajo para lograr una certificación. (Toro, 2014, p. 28).

Del mismo modo la metodología RUP, consta de 4 fases o etapas, que son: fase de inicio, fase de elaboración, fase de construcción y la fase de transición (Granados, 2015).

La metodología XP es considerada una metodología ágil y está clasificada como un conjunto de buenas prácticas que se viene perfeccionando por parte de la comunidad de los developers, y que tiene como propósito dar soluciones a los diferentes inconvenientes que puedan ocurrir en la entrega del software, de manera rápida y confiable (Laínez, 2014, p. 116).

La metodología Scrum es aquel proceso donde se aplican de manera regular diversos procesos para poder trabajar en equipo, y así obtener el mejor resultado de un proyecto (Torras, 2015).

La metodología Scrum se caracteriza por tener equipos autónomos y autogestionados que distribuyen su conocimiento de forma abierta, es vista como una habilidad de mejora incremental y a la vez basar la calidad del resultado en la comprensión implícita de las personas y su creatividad; no como otras metodologías que realizan una planificación completa del producto (Palacio, 2020).

Así mismo, en esta investigación se empleó la validación de expertos para seleccionar la metodología de desarrollo (ver Anexo 12). En la siguiente tabla se puede observar los puntajes y resultados:

Tabla N°1: Validación de la metodología de desarrollo de software

<b>Experto</b>	<b>Grando</b>	<b>SCRUM</b>	<b>RUP</b>	<b>XP</b>
Villaverde Medrago, Hugo	Doctor	33	28	23
Fermín Pérez, Félix Armando	Magister	38	30	27
Johnson Romero, Guillermo Miguel	Magister	36	29	28
<b>Total</b>		<b>107</b>	<b>87</b>	<b>78</b>

Como se observa en la tabla N°1, se visualiza que la metodología de desarrollo con más alta puntuación fue Scrum, por ello, es la que se empleó en esta investigación.

### **III. MÉTODO**

### **3.1 Tipo y diseño de investigación**

La investigación que se usó fue de tipo aplica, porque va a permitir constituir la relación de causa entre el sistema web y el control de inventario en la empresa. De esta manera, se podrá manipular la variable dependiente que en este caso es el control de inventario, en condiciones estrictamente registradas, con el objetivo de poder describir de qué manera o por qué razón se genera un escenario o acontecimiento cuando se implemente el sistema web.

Este tipo de investigación también es llamada activa, dinámica o práctica, y se encuentra directamente ligado a la llamada investigación básica, debido a que es dependiente de hallazgos y aportes teóricos para poder implementar algún tipo de solución al problema que se planteó (Valderrama, 2014).

Para esta investigación se utilizaran una variedad de datos para el analisis, donde se evaluara un antes y un despues en relación a la causa y efecto de las variables, teniendo como un antes al proceso de control de inventario y un después ya contando con el sistema web en dicho proceso, con el fin de optimizar las actividades del mismo.

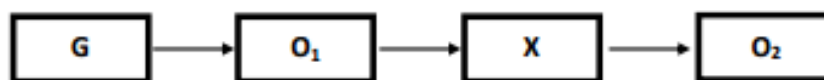
En el momento en que un investigador intenta establecer un potencial efecto de una causa que es manipulable y en estas circunstancias es cuando se utilizan los diseños experimentales (Hernández, 2014, p. 130).

Existen pasos para realizar un diseño experimental y son: aplicar un pre test, esto sirve para medir la variable dependiente, aplicar la variable independiente y al final se aplica un post test para realizar nuevamente la medida de la variable dependiente. Al realizar un diseño de estudio pre-experimental existe una subclase descrita como diseño de la prueba previa y posterior a un grupo, en estos diseños se tiene que medir dos veces a un determinado grupo de unidades de prueba, puesto que no hay un grupo de control. Lo primero es tomar la medición previa a realizar el tratamiento ( $O_1$ ), luego el grupo seleccionado se expone al tratamiento (X) y por último se toma nuevamente una medición ( $O_2$ ) (Hernández, 2014, p. 140).

Por lo tanto, el esquema que se utilizara para la investigación fue de tipo experimental, de las cuales se realizará un pre-experimental. Se utilizará el diseño pre-experimental y para ello primero se tiene que establecer una diferencia entre O<sub>1</sub> y O<sub>2</sub>, donde O<sub>1</sub> fue el proceso de control de inventario antes de la aplicación de un sistema web y el O<sub>2</sub> fue el proceso de control de inventario después de la aplicación del sistema web, esto se realiza para verificar si tuvo una influencia positiva o negativo en el proceso. (ver figura N°3)

En esta investigación se elaborara un pre – test y un post – test, por lo que se requiriera de un análisis detallado y específico. Se analizara en primera instancia el resultado del control de inventario en la empresa.

*Figura N°3. Diseño de estudio*



Fuente: Hernández et al. (2014).

Donde:

**G:** Grupo experimental, en otras palabras es la población que se tomo para realizar el estudio en la que se realizarán las pruebas de Pre y Post.

**X:** Es la ejecución de la aplicación web.

**O<sub>1</sub>:** Resultados de las pruebas sin el sistema web ejecutado.

**O<sub>2</sub>:** Resultados de las pruebas con el sistema web ejecutado.

Los resultados se plasmaran luego de realizar la medición correspondiente donde se establecera la diferencia entre la variable O<sub>1</sub> y O<sub>2</sub> y de esta manera comprobar si hubo una influencia en la productividad en los procesos para posteriormente generar la validación con las hipótesis planteadas.

### **3.2 Variable y operacionalización**

La variable independiente, sistema web, es una herramienta que permite el registro, seguimiento y salida de los productos de la empresa Selva Gas Tocache SAC, con el fin de mejorar el control de inventario, por lo que actualmente

trabajan de manera semiautomatizada utilizando hojas de cálculo, cuadernos de registro de ingresos y egresos causándoles malestar en el procesamiento de información.

La variable dependiente, proceso de control de inventario, está orientado a la gestión de recibimiento, stock y movimientos de las diferentes mercancías que se encuentran en la empresa Selva Gas Tocache S.A.C.

Asimismo, para la medición de la variable dependiente se utilizó las dimensiones Almacén y Entrega.

Para la dimensión de Almacén se utilizó el índice de rotación de stock como indicador y para la dimensión de Entrega se usó el nivel de servicio como indicador, y como la escala de medición se utilizó la Razón.

### **3.3 Población, muestra y muestreo**

En este trabajo de investigación, la población está basada en el promedio de ventas al día por 30 días del total de un mes, es decir  $20 * 30 = 600$  productos, para el índice de rotación, y para el nivel de servicio se tomó la cantidad promedio de pedidos que obtiene la empresa al mes es decir 540.

La población es el conjunto de la anomalía a experimentar, donde los sujetos tienen una característica común la cual se trabaja y da origen a los datos de la investigación (Hernández, 2014, p. 170).

La muestra es una porción representativa de un todo es decir de una población o universo, debido a que refleja las mismas características de la población al momento de realizar una técnica adecuada de muestreo de la cual proviene, además tiene que estar representado por un número mínimo y óptimo de unidades, el cual se determina aplicando procedimientos diversos (Valderrama, 2014, p. 182).

### **3.4 Técnicas e instrumentos de recolección de datos**

La técnica del fichaje es utilizada para el trabajo intelectual y a su vez facilita la ordenación de ideas, la sistematización de bibliografías y el trabajo de síntesis (Mingrone, 2007).

El fichaje es un instrumento propio de aprendizaje que permite plasmar textualmente ideas haciendo uso de figuras metodológicas, tales como el resumen, un esquema, la síntesis, etc. La ficha beneficia a la captación, integración y sistematización de ideas que a su vez permite tener una mejor comprensión de la realizar (Migrone, 2007).

Tabla N°2: *Instrumento para obtener los datos*

<b>Variable</b>	<b>INDICADOR</b>	<b>TÉCNICA</b>	<b>INSTRUMENTOS</b>
Variable dependiente: Proceso de Control de Inventario	Nivel de servicio	Fichaje	Ficha de Registro (ver Anexo 5)
	Índice de rotación de Stock	Fichaje	Ficha de Registro (ver Anexo 5)

En este estudio también se realizó una validez de expertos para evaluar el instrumento de medición. A continuación, se muestran los puntajes dados por los 3 expertos.

De acuerdo a la puntuación que se muestra en la tabla N°3 y en el anexo N°7, sobre el instrumento del indicador Nivel de servicio, se obtiene que, se adquirió una puntuación promedio de 81%, el cual muestra un grado de confianza aceptable. Por lo tanto, dicho instrumento, fue adecuado para el estudio.

Tabla N°3: *Validez del instrumento de nivel de servicio*

<b>N°</b>	<b>Expertos</b>	<b>Grado Académico</b>	<b>Puntaje</b>
1	Villaverde Medrano, Hugo	Doctor	75%
2	Fermín Pérez, Félix Armando	Magister	84%
3	Johnson Romero, Guillermo Miguel	Magister	83%

Asimismo, como se corrobora en la tabla N°4 y en el anexo N°7, sobre el instrumento del indicador Índice de rotación de stock, se obtiene que, se adquirió una puntuación promedio de 80%, el cual muestra un grado de confianza aceptable. Por lo tanto, dicho instrumento, fue adecuado para el estudio.



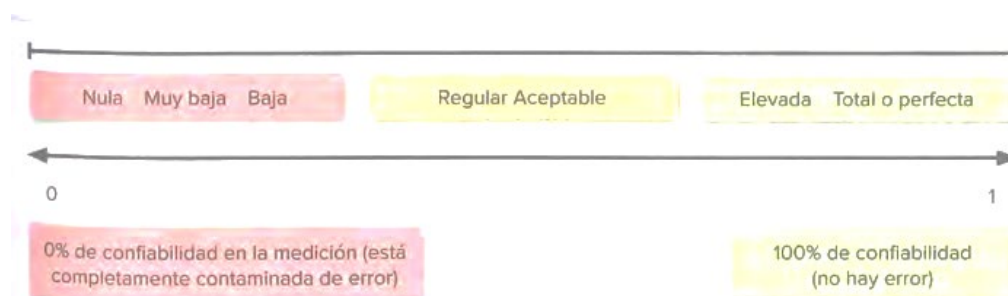
Tabla N°4: Validez del instrumento del índice de rotación de stock

N°	Expertos	Grado Académico	Puntaje
1	Villaverde Medrano, Hugo	Doctor	75%
2	Fermín Pérez, Félix Armando	Magister	84%
3	Johnson Romero, Guillermo Miguel	Magister	81%

Probar que el instrumento es confiable nos da como resultado, datos sólidos e íntegros. De la misma manera, los autores opinan lo siguiente:

Para establecer la confiabilidad o fiabilidad de un instrumento de medida existen diversos procedimientos para realizar su cálculo y todas esas fórmulas oscilan entre el 0 y 1, donde el valor de 0 significa que su confiabilidad es nula y al contrario el valor 1 representa el máximo de confiabilidad. En otras palabras, mientras más se acerque al valor 0 mayor será el error que brindará la medición (Hernández, 2010, 238). (ver figura N°4)

Figura N°4. Interpretación de un coeficiente de confiabilidad

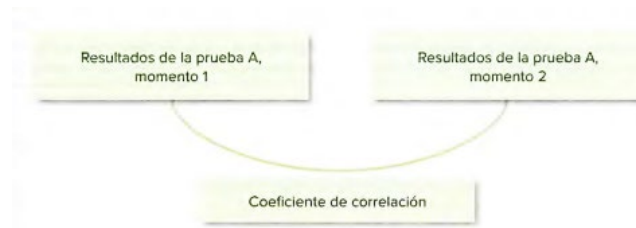


Fuente: Hernández et al. (2010), p.239

Uno de los tantos procedimientos que se utilizan para comprobar la confiabilidad a través de un coeficiente es la medida de estabilidad.

En estos procedimientos, dicha herramienta de medida se debe realizar dos o más veces al mismo conjunto de individuos, cada cierto tiempo. Si la reciprocidad entre las distintas aplicaciones sigue siendo muy positiva, se considera que es un instrumento confiable (Hernández, 2010, 222).

Figura N°5. Medida de estabilidad



Fuente: Hernández et al. (2010), p.223

### 3.5 Procedimientos

El procedimiento para obtener y hacer uso de la información se desarrolló de la siguiente manera:

Primero, se realizó una llamada a la empresa Selva Gas Tocache S.A.C para acordar una cita con la gerente general Patricia Lesly Estupiñan Palma para poder realizar la entrevista vía zoom.

Luego, cuando llegó el día de la cita, se le realizó la entrevista a la gerente general (ver anexo N°9) y aceptó que la tesis titulada: “Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C”, se realice.

Más adelante, se acordó otra cita para poder obtener los datos, para luego registrarlos en las fichas de registro, de los indicadores Nivel de servicio (ver anexo N°5) y Índice de rotación de stock (ver anexo N°5).

### 3.6 Método de análisis de datos

Los datos obtenidos mediante los instrumentos de medición fueron procesados mediante la estadística, y en base a ello se comprobaron las hipótesis de estudio.

Teniendo en cuenta la prueba de normalidad que es utilizada para comprobar si los resultados obtenidos del estudio cumplen con una distribución normal, para ello en este estudio se empleó la prueba de Shapiro – Wilks, al igual que otras pruebas de normalidad verifica si los datos se comportan de manera normal y como la muestra es menor a 50, es aplicable dicha prueba.

A continuación se muestra la hipótesis general:

### **Hipótesis Nula**

H<sub>0</sub>: El sistema web no mejora el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

### **Hipótesis Alternativa**

H<sub>a</sub>: El sistema web mejora el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

A continuación se muestra las hipótesis específicas:

Hipótesis específica 1: El sistema web mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

NS<sub>a</sub>: Nivel de servicio antes de la implementación del sistema web.

NS<sub>d</sub>: Nivel de servicio después de la implementación del sistema web.

### **Hipótesis Nula**

H<sub>0</sub>: El sistema web no mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_0: NS_a \geq NS_d$$

### **Hipótesis Alternativa**

H<sub>a</sub>: El sistema web mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_a: NS_a < NS_d$$

Hipótesis específica 2: El sistema web mejora el índice de rotación stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

IRS<sub>a</sub>: Índice de rotación de stock antes de la implementación del sistema web.

IRS<sub>d</sub>: Índice de rotación de stock después de la implementación del sistema web.

### **Hipótesis Nula**

H<sub>0</sub>: El sistema web no mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_0: IRS_a \geq IRS_d$$

### **Hipótesis Alterna**

H<sub>a</sub>: El sistema web mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_0: IRS_a < IRS_d$$

### **3.7 Aspectos éticos**

El autor se comprometió a que la información brindada por la empresa Selva Gas Tocache S.A.C no fuera alterada y que solamente será utilizada para el desarrollo de esta investigación. Por otro lado, los datos mas críticos de la empresa no serán divulgados para respetar la confidencialidad de los mismo.

Al mismo tiempo, todo resultado del estudio no sufrieron modificaciones y toda aquella información no perteneciente al investigador, fueron citados y referenciados, de esta manera poder respetar los derechos de autor y evitando cualquier tipo de plagio.

## **IV. RESULTADOS**

## Descripción

El estudio se ejecutó en dos fases para comprobar el rechazo o afirmación de la hipótesis a través del diseño Pre – Experimental. En la primera fase se realizó el Pre – Test, que consistió en obtener una medición a través de las fórmulas obtenidas para cada indicador, esta medición se realizó antes de la implementación del software. Luego, en la segunda fase se realizó el Post – Test, que también consistió en obtener una medición a través de las fórmulas obtenidas para cada indicador, esta medición se realizó después de la implementación del software. En consecuencia, estos datos obtenidos de cada fase se pueden comparar y verificar si hubo una mejora o no.

## Análisis descriptivo

### Indicador: Nivel de servicio.

La tabla N°5 mostró que el indicador de nivel de servicio tuvo un promedio de 76.23% en el Pre – Test, y en el Post – Test tuvo un promedio de 90.83%, Por lo tanto, hubo una variación positiva en los resultados luego de la implementación del sistema web.

Además, se puede apreciar que el indicador nivel de servicio tuvo un mínimo de 58% y después tuvo un mínimo de 80% con la implementación del sistema web, asimismo tuvo un máximo de 89% y después tuvo un máximo de 100% con la implementación del sistema web.

Referente a la desviación, el indicador nivel de servicio en el Pre – Test tuvo una variabilidad de 6.92, y en el Post – Test tuvo un valor de 5.65.

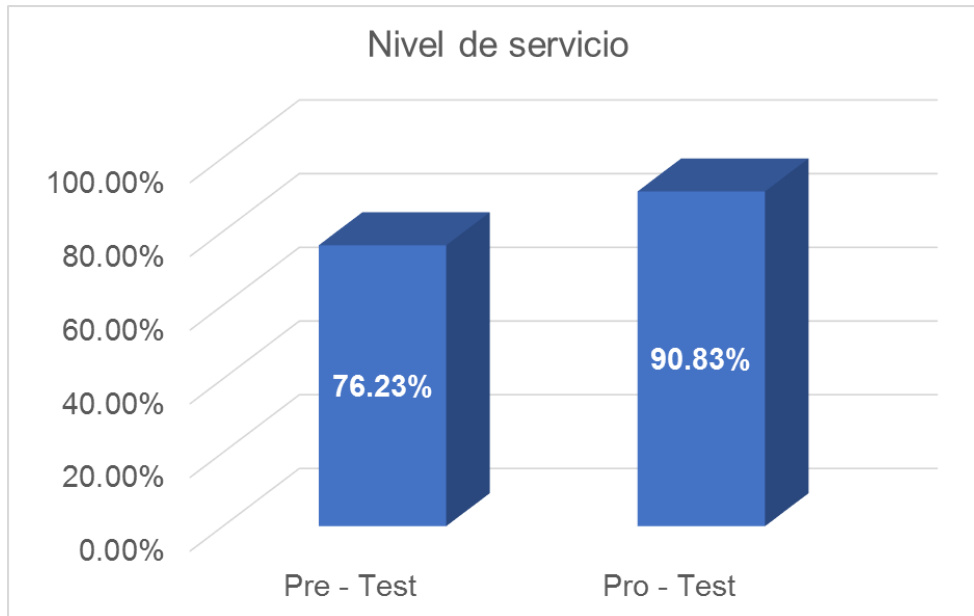
Tabla N°5: Análisis descriptivo del nivel de servicio

	N	Mínimo	Máximo	Media	Desv. típ.
Pre_Test	30	58,00	89,00	76,2333	6,92165
Post_Test	30	80,00	100,00	90,8333	5,65736
N válido (según lista)	30				

Como se puede observar en la figura N°6 el indicador nivel de servicio antes de la implementación del sistema tuvo un valor de 76.23%, pero luego de

implementar el sistema se obtuvo un valor de 90.83%, por lo cual, se concluye que el servicio que se le brinda al cliente mejoró en un 14.6%.

*Figura N°6. Indicador nivel de servicio antes y después de implementar el sistema*



**Indicador: Índice de rotación de stock.**

La tabla N°6 mostró que el indicador índice de rotación de stock tuvo un promedio de 24.2% en el Pre – Test, y en el Post – Test tuvo un promedio de 42.83%, Por lo tanto, hubo una variación positiva en los resultados luego de la implementación del sistema web.

Además, cabe mencionar que el indicador índice de rotación de stock tuvo un mínimo de 13% y después tuvo un mínimo de 17% con la implementación del sistema web, asimismo tuvo un máximo de 62% y después tuvo un máximo de 91% con la implementación del sistema web.

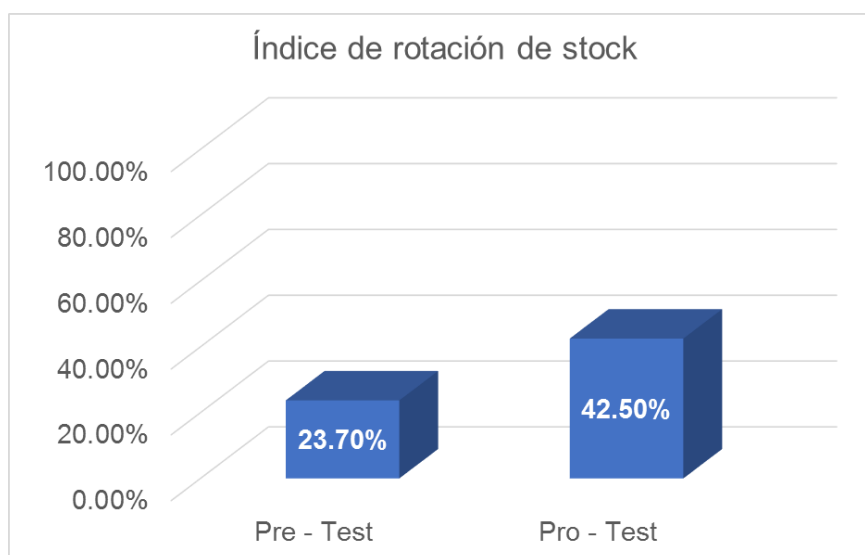
Con respecto a la desviación, el indicador índice de rotación de stock en el Pre – Test tuvo una variabilidad de 10.6, y en el Post – Test tuvo un valor de 26.05.

Tabla N°6: *Análisis descriptivo del índice de rotación*

Estadísticos descriptivos					
	N	Mínimo	Máximo	Media	Desv. típ.
Pre_Test	30	12,00	45,00	23,6667	8,98402
Post_Test	30	17,00	92,00	42,5000	25,55420
N válido (según lista)	30				

Se puede observar en la figura N°7 que el indicador índice de rotación de stock antes de la implementación del sistema tuvo un valor de 23.7%, pero luego de implementar el sistema obtuvo un valor de 42.5%, por lo cual, se concluye que la renovación de los productos mejoró en un 18.8%.

Figura N°7. *Indicador índice de rotación de stock antes y después de implementar el sistema*





## Análisis inferencial

### Indicador: Nivel de servicio.

En la tabla N°7 se observa que el sig. en el Pre – Test fue de 0,245 y como es mayor a 0.05 se distribuye de manera normal. Ahora, en el Post – Test el sig. fue de 0.001 lo que significa que no se distribuye de una manera normal. Al no tener una distribución normal, la prueba estadística que se aplicó fue Wilcoxon.

Tabla N°7: Prueba de Shapiro-Wilk del indicador de nivel de servicio

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Test	,956	30	,245
Post_Test	,852	30	,001

### Prueba de Hipótesis 1

Hipótesis específica 1: El sistema web mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

Indicador: Nivel de servicio.

Hipótesis estadísticas

Definición de variables:

NS<sub>a</sub>: Nivel de servicio antes de la implementación del sistema web.

NS<sub>d</sub>: Nivel de servicio después de la implementación del sistema web.

H<sub>0</sub>: El sistema web no mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_0: NS_a \geq NS_d$$

H<sub>a</sub>: El sistema web mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_a: NS_a < NS_d$$

Para verificar si la hipótesis se acepta o se rechaza se optó por utilizar la prueba de Wilcoxon ya que los datos proporcionados por el indicador nivel de servicio no se distribuyen de manera normal. Los resultados se visualizan en las tablas N°8.

Tabla N°8: Prueba de Wilcoxon del indicador nivel de servicio

		Rangos		
		N	Rango promedio	Suma de rangos
Post_Test - Pre_Test	Rangos negativos	30 <sup>a</sup>	15,50	465,00
	Rangos positivos	b	,00	,00
	Empates	0 <sup>c</sup>		
	Total	30		

Se observa en la tabla N°9 que el valor de Significancia estadística es menor a 0.05, por esta razón se rechaza la hipótesis nula y se acepta la alterna.

Tabla N°9: Estadísticos de prueba

	Post_Test - Pre_Test
Z	-4,787 <sup>b</sup>
Sig. asintót. (bilateral)	,000

a. Prueba de los rangos con signo de Wilcoxon

b. Basado en los rangos negativos.

### Indicador: Índice de rotación de stock.

En la tabla N°10 se observa que el sig. en el Pre – Test fue de 0,062 y como es mayor a 0.05 se distribuye de manera normal. Ahora, en el Post – Test el sig. fue de 0.000 lo que significa que no se distribuye de una manera normal. Al no tener una distribución normal, la prueba estadística que se aplicó fue Wilcoxon.

Tabla N°10: Prueba de Shapiro-Wilk del indicador índice de rotación de stock

	Shapiro-Wilk		
	Estadístico	gl	Sig.
Pre_Test	,934	30	,062
Post_Test	,802	30	,000

## Prueba de Hipótesis 2

Hipótesis específica 2: El sistema web mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

Indicador: Índice de rotación de stock.

Hipótesis estadísticas

Definición de variables.

IRS<sub>a</sub>: Índice de rotación de stock antes de la implementación del sistema web.

IRS<sub>d</sub>: Índice de rotación de stock después de la implementación del sistema web.

H<sub>0</sub>: El sistema web no mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_0: IRS_a \geq IRS_d$$

H<sub>a</sub>: El sistema web mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

$$H_a: IRS_a < IRS_d$$

Para verificar si la hipótesis se acepta o se rechaza se optó por utilizar la prueba de Wilcoxon ya que los datos proporcionados por el indicador índice de rotación de stock no se distribuyen de manera normal. Los resultados se visualizan en las tablas N°11.

Tabla N°11: Prueba de Wilcoxon del indicador índice de rotación de stock

		Rangos		
		N	Rango promedio	Suma de rangos
Post_Test - Pre_Test	Rangos negativos	0 <sup>a</sup>	,00	,00
	Rangos positivos	30 <sup>b</sup>	15,50	465,00
	Empates	0 <sup>c</sup>		
	Total	30		

Se observa en la tabla N°12 que el valor de Significancia estadística es menor a 0.05, por esta razón se rechaza la hipótesis nula y se acepta la alterna.

Tabla N°12: Estadísticos de prueba

	Post_Test - Pre_Test
Z	-4,786 <sup>b</sup>
Sig. asintót. (bilateral)	,000

a. Prueba de los rangos con signo de Wilcoxon

b. Basado en los rangos negativos.

## **V. DISCUSIÓN**

El presente trabajo de investigación “Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.”, se infiere que, en base a los datos recolectados y posteriormente procesados mediante la herramienta estadística SPSS, se afirma la hipótesis general en la cual se establece que el sistema web mejora el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C.

Debido a que la distribución de los datos proporcionados por los indicadores no se distribuyó de forma normal se empleó la prueba estadística Wilcoxon, los datos generados por dicha prueba corroboran la aceptación de la hipótesis alternativa y el rechazo de la hipótesis nula. Puesto que los datos obtenidos por el indicador nivel de servicio indica un incremento del 14.6% y el indicador índice de rotación de stock indica un incremento del 18.63%.

Los resultados que se obtuvieron demuestran que la implementación del sistema web mejoró el proceso de control de inventario, lo cual se expresa al comparar los datos obtenidos antes y después de la implementación del sistema. En el primer indicador nivel de servicio se tuvo en su pre – test un 76.23%, sobre una muestra de 30 días hábiles, posteriormente se tuvo en su post – test un 90.83% lo cual se interpreta como un incremento positivo de 14.6%, asimismo en el segundo indicador índice de rotación de stock se tuvo en su pre – test un 23.7%, teniendo la misma muestra, posteriormente se tuvo en su post – test un 42.5% que se define como un incremento positivo de 18.8%.

Asimismo, en la investigación de Hernandez (2019) en la Municipalidad Provincial de San Miguel – Cajamarca, tuvo como problemática que el sistema actual era muy antiguo e inutilizable para la institución. La implementación de un sistema web como solución a dicho problema ayudo a mejorar la usabilidad de los recursos tecnológicos, un mejor rendimiento de los usuarios, mayor rapidez en el registro de la información o generación de reportes.

## **VI. CONCLUSIONES**

La presente investigación llevo a las siguientes conclusiones:

Se muestra que mediante la creación y utilización del sistema web el proceso de control de inventario de la empresa Selva Gas Tocache S.A.C tuvo una mejoría mayor al 15%, puesto que hubo una influencia positiva del sistema en el indicador de nivel de servicio y en el indicador índice de rotación de stock, lo cual demuestra el cumplimiento de los objetivos de la investigación.

Se evidencia que hubo un aumento en el indicador nivel de servicio de un 14.6% luego de hacer uso del sistema web, ya que dicho sistema ayudo a mejorar el servicio que brindaba la empresa hacia el cliente, respondiendo de manera más rápida las consultas de los clientes referente a la compra de sus productos.

Se muestra que hubo un aumento en el indicador índice de rotación de stock de un 18.8% luego de hacer uso del sistema web, ya que dicho sistema incremento la renovación de los productos, y como consecuencia se generan más ventas y una mejor gestión de las existencias.



## **VII. RECOMENDACIONES**

Se recomienda utilizar los mismos indicadores que se utilizó en esta investigación ya que, si no se lleva un buen control de la rotación de las existencias y un buen servicio hacia los clientes, no se tendrá un rendimiento adecuado y una buena gestión en el control del inventario.

Se recomienda expandir el almacén de los productos debido al gran crecimiento de la demanda, lo que produce faltantes de productos antes de la llegada de los pedidos realizados a los proveedores.

Se recomienda verificar semanalmente que producto tuvo más ventas y de esta manera dar prioridad a dicho producto al momento de realizar el pedido a los proveedores.

Se recomienda añadir un diseño responsive a los módulos de usuarios, productos, entrada y salida de productos del sistema web para que de esta manera se pueda acceder desde un dispositivo móvil y no tener inconvenientes con el servicio en caso de que el ordenador este fuera de servicio por problemas técnicos.

## **REFERENCIAS**

ARENAL, Carmen. Gestión de inventarios: UF0476. 2020 [fecha de consulta: 20 de abril de 2021]. Consultado en <https://elibro.net/es/ereader/bibliourp/126745>

AYMERICH, David. Gestión logística y comercial. Primera edición. Mc Graw Hill. 2018 [fecha de consulta: 30 de abril de 2021]. Consultado en <https://www.mheducation.es/bcv/guide/capitulo/8448199316.pdf>

CANTURIAS, F. Locales de venta de balones de gas se triplicó en nueve años: solo tres regiones tienen cobertura inferior a 30%. 2020 [fecha de consulta: 16 de abril de 2021]. Consultado en <https://gestion.pe/economia/locales-de-venta-de-balones-de-gas-se-triplico-en-nueve-anos-solo-tres-regiones-tienen-cobertura-inferior-a-30-noticia/>

CARDON, José, OREJUELA, Juan, y ROJAS, Carlos. Gestión de inventario y almacenamiento en el sector de alimentos concentrados. Revista EIA [en línea]. Diciembre 2018, n.º 15. [fecha de consulta: 18 de abril de 2021]. Disponible en <https://www.doi.org/10.24050/reia.v15i30.1066>. ISSN: 1794-123

CARRILLO, S, BRAVO , L, y CÓRDOVA, Z. El control de los inventarios y sus implicancias financieras en el capital de trabajo de las empresas. Revista de Economía, Finanzas y Negocios 2016, Vol. 06(Num. 01), 13 - 21 pág.

CELINA, Heidi y CAMPO, Adalberto. Metodología de investigación y lectura crítica de estudios. [en línea] Revista Colombiana de Psiquiatría 2006, XXXIV, 572 - 580 pág. [fecha de consulta: 15 de abril de 2021]. Disponible en [http://www.scielo.org.co/scielo.php?script=sci\\_arttext&pid=S0034-74502005000400009&lang=en](http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0034-74502005000400009&lang=en)

FOLLEGATI, Carla. El economista América Perú. 2019 [fecha de consulta: 10 de abril de 2021]. Disponible en <https://www.eleconomistaamerica.pe/empresas-eAm-peru/noticias/10079964/09/19/PYMES-cuatro-beneficios-de-automatizar-las-ventas-.html>

FRESNEDA, Jorge. Pasos en un proceso de gestión de inventarios y beneficios para la empresa. 2019 [fecha de consulta: 20 de abril de 2021]. Disponible en <https://revistadigital.inesem.es/gestion-empresarial/el-proceso-de-gestion-de-inventarios/>

GRANADOS, Rafael. Desarrollo de aplicaciones web en el entorno servidor. Málaga: IC Editorial, 2014. 128p. ISBN: 9788416433063

Grupo de desarrollo global de PostgreSQL. (2021). Documentación de PostgreSQL 13.2. 2021 [fecha de consulta: 25 de abril de 2021]. Disponible en <https://www.postgresql.org/docs/current/intro-what-is.html>

HERNANDEZ, Jhordan. Implementación de un sistema web de control de inventarios y su influencia para controlar y manipular los bienes patrimoniales de la Municipalidad Provincial de San Miguel - Cajamarca. 2019 [fecha de consulta: 21 de abril de 2021]. Disponible en <https://repositorio.unc.edu.pe/handle/UNC/2737>

HERNÁNDEZ, Roberto, FERNÁNDEZ, Carlos, y BAPTISTA, María. Metodología de la investigación, Quinta edición. México: The McGraw-Hill Companies, Inc. 2010 [fecha de consulta: 01 de mayo de 2021]. Disponible en <https://www.icmujeres.gob.mx/wp-content/uploads/2020/05/Sampieri.Met.Inv.pdf>

HERNÁNDEZ, Roberto, FERNÁNDEZ, Carlos, y BAPTIST, María. Metodología de la investigación Sexta edición. Mc Graw Hill Education, INTERAMERICANA EDITORES, S.A. 2014 [fecha de consulta: 13 de abril de 2021]. Disponible en <https://www.uca.ac.cr/wp-content/uploads/2017/10/Investigacion.pdf>

INEI. Informe Técnico: Producción nacional. 2021 [fecha de consulta: 25 de febrero de 2021]. Disponible en <https://www.inei.gob.pe/media/MenuRecursivo/boletines/03-informe-tecnico-produccion-nacional-ene-2021.pdf>

JIMÉNEZ, Javier. Aplicaciones web. Macmillan Iberia, S.A. 2013 [fecha de consulta: 25 de abril de 2021]. Disponible en <https://elibro.net/es/ereader/bibliourp/43262?page=8>

LAÍNEZ, José. Desarrollo de software ágil: extreme programming y scrum. Madrid: CreateSpace Independent Publishing Platform, 2014. 116p. ISBN: 9781502952226

LUCAS, Kerly. Desarrollo e Implementación de Aplicación web para el control de inventario del local comercial Máquinas Hidalgo. 2017 [fecha de consulta: 28 abril de 2021]. Disponible en <https://dspace.ups.edu.ec/bitstream/123456789/15097/1/UPS-GT002054.pdf>

LOZANO, Héctor. Análisis y desarrollo de un sistema web para la gestión kardex de un almacén. 2017 [fecha de consulta: 27 de abril de 2021]. Disponible en [http://oa.upm.es/48282/11/TFM\\_HECTOR\\_JAHIR\\_LOZANO\\_BRIONES.pdf](http://oa.upm.es/48282/11/TFM_HECTOR_JAHIR_LOZANO_BRIONES.pdf)

MARTIN, Esneider. Diseño e implementación de sistema de inventarios para el almacén de pinturas y ferretería Ferrecolor. 2019 [fecha de consulta: 24 de abril de 2021]. Disponible en <https://repository.ucc.edu.co/handle/20.500.12494/8557>

MENDOZA, G. Grupo El Comercio : Gestion.pe. 2019 [fecha de consulta: 11 de abril de 2021]. Disponible en <https://gestion.pe/economia/empresas/empresas-elevan-ventas-25-automatizar-gestion-inventarios-272267-noticia/>

MICROSOFT. Documentación técnica de SQL Server. 2017 [fecha de consulta: 30 de abril de 2021]. Disponible en <https://docs.microsoft.com/es-es/sql/relational-databases/databases/databases?view=sql-server-ver15>

MINGRONE, Patricia. Metodología Del Estudio Eficaz / Methodology of Effective Study: Como Estudiar? Como Aprender?/ How to Study?. 2007 Editorial Bonum. ISBN: 9789505077366

LOTFY, Mohamed y PYATT, Kevin. The Mean Stack Web Application Development Platform: Tutorial Presentation. Revista de Ciencias de la Computación en Universidades [en línea]. Diciembre 2018 n.º 34. [fecha de consulta: 22 de abril de 2021]. Disponible en <https://dl.acm.org/doi/10.5555/3282588.3282602> ISSN: 1937-4763

MONTE, Josep. Implantar scrum con éxito. Editorial UOC. 2016 [fecha de consulta: 20 de abril de 2021]. Disponible en <https://elibro.net/es/ereader/bibliourp/58575>

MONTEZA, Celinda. Diseño e implementación de un sistema web para la mejora de procesos en la gestión de almacén de la empresa Carrocería Lima Traylers

S.A.C. 2019 [fecha de consulta: 23 de abril de 2021]. Disponible en <http://repositorio.ulasamericas.edu.pe/handle/upa/599>

NOLASCO, Jorge. Python: aplicaciones prácticas. Editorial RA-MA. 2018 [fecha de consulta: 20 de abril de 2021]. Disponible en <https://elibro.net/es/ereader/bibliourp/106523?page=14>

ORTEGA, José. Seguridad en aplicaciones Web Java. Editorial RA-MA. 2018 [fecha de consulta: 20 de abril de 2021]. Disponible en <https://elibro.net/es/lc/bibliourp/titulos/106511>

OTWELL, Taylor. Laravel Documentation - Laravel - The PHP Framework. 2011 [fecha de consulta: 20 de abril de 2021]. Disponible en <https://laravel.com/docs/8.x>

PALACIO, Marta. Scrum Master. Scrum Manager. 2020 [fecha de consulta: 30 de abril de 2021]. Disponible en [https://scrummanager.net/files/scrum\\_master.pdf](https://scrummanager.net/files/scrum_master.pdf)

PAVÓN, Jacobo y LLARENA, Ezequiel. Creación de un sitio web con PHP y MySQL. 2015 [fecha de consulta: 10 de abril de 2021]. Disponible en <https://elibro.net/es/ereader/bibliourp/106491?page=17>

PHP GROUP. PHP. 2021 [fecha de consulta: 02 de abril de 2021]. Disponible en <https://www.php.net/manual/es/intro-what-is.php>

PURISACA, Gerson y ZAVALA, Rossen. Sistema web para el control de inventario del área de gabinete en el proyecto del museo de sitio de Túcume-Lambayeque. Lambayeque, Perú: Universidad Nacional de Trujillo. 2019 [fecha de consulta: 22 de abril de 2021]. Disponible en <https://dspace.unitru.edu.pe/handle/UNITRU/13931>

REAL ACADEMIA ESPAÑOLA (RAE). Plataforma de recursos lingüísticos de la RAE. 2021 [fecha de consulta: 25 de abril de 2021]. Disponible en <http://dle.rae.es/?id=M2v6jgO>

RINCÓN, Carlos y VILLARREAL, Fernando. Costos decisiones empresariales. Bogotá: Ecoe Ediciones, 2013. 65p. ISBN: 9789586486163

SAMANIEGO, Hernán. Un modelo para el control de inventarios utilizando dinámica de sistemas. Ecuador: Estudios de la Gestión, 2019. 137p. ISSN: 2550-6641

TORRAS, Ivo. Scrum. 2015 [fecha de consulta: 29 de abril de 2021]. Disponible en <https://metodologiascrum.readthedocs.io/en/latest/Scrum.html>

TORO, Francisco. Administración de proyectos de informática. Bogotá: Ecoe Ediciones, 2014. 28p. ISBN: 9789586488167

VALDERRAMA, Santiago. Pasos para elaborar proyectos de Investigación Científica Cuantitativa, Cualitativa y Mixta. 2014 Editorial San Marcos. ISBN: 9786123028787

VERA, Christian. Desarrollo e implementación de un sistema web para el control de inventario y alquiler de maquinarias de la empresa Megarent S.A. 2019 Ecuador: Universidad Politécnica Salesiana Sede Guayaquil. [fecha de consulta: 15 de abril de 2021]. Disponible en <https://dspace.ups.edu.ec/bitstream/123456789/17525/1/UPS-GT002706.pdf>

VIDAL, Carlos. Fundamentos de control y gestión de inventarios. Programa Editorial Universidad del Valle, 2017. ISBN: 9789586708630

WESTREICHER, Guillermo. Control de inventario. 2020 [fecha de consulta: 26 de abril de 2021]. Disponible en <https://economipedia.com/definiciones/control-de-inventario.html>

YUPANQUI, María. Desarrollo de un sistema web para la facturación de la lavadora y lubricadora "Los Angeles" aplicando el Framework Laravel. Riobamba. 2019 [fecha de consulta: 12 de abril de 2021]. Disponible en <http://dspace.esepoch.edu.ec/bitstream/123456789/12256/1/18T00791.pdf>



## **ANEXOS**

### Anexo N°1: Matriz de operacionalización de variables

Variable	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Escala de medición
<b>Variable independiente (VI):</b> Sistema web	Una aplicación web es un programa de software cliente-servidor donde el back-end se ejecuta en un servidor web. La parte del front-end, que incluye la interfaz de usuario y la lógica del lado del cliente, se ejecuta dentro de un navegador web (Lotfy y Pyatt, 2018, p. 99).	El sistema web permite incrementar el rendimiento en el proceso de control de inventario, debido a la facilidad y rapidez en que se realizar los procesos.			
<b>Variable dependiente (VD):</b> Proceso de control de inventario	El control de inventario es uno de los tantos procesos que tiene una empresa, el cual se encarga de administrar la información sobre las diferentes mercancías que existan en el almacén. Su principal función es compilar información de las entrada y salidas de los productos (Westreicher, 2020).	Generación de reportes, ingreso, seguimiento y salida de los productos	Entrega	- Nivel de servicio	Razón (Cuantitativo)
			Almacén	- Índice de rotación de Stock	Razón (Cuantitativo)

## Anexo N°2: Matriz de consistencia

Título: Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C

Problemas	Objetivos	Hipótesis	Variables	Variable Dependiente				Métodos
<p>General:</p> <p>¿De qué manera influye un sistema web en el proceso de control de inventario en la empresa Selva Gas Tocache SAC?</p>	<p>General:</p> <p>Determinar la influencia del sistema web en el proceso de control de inventario de la empresa Selva Gas Tocache SAC.</p>	<p>General:</p> <p>El sistema web mejora el proceso control de inventario de la empresa Selva Gas Tocache SAC.</p>	<p>Independiente:</p> <p>Sistema Web</p>	Operacionalización de variable				<p><b>Tipo de investigación:</b></p> <p>Aplicada</p> <p><b>Diseño de investigación:</b></p> <p>Experimental Pre-Experimental</p>
<p>Secundarios:</p> <p>¿De qué manera influye un sistema web en el nivel de servicio para el proceso de control de inventario en la empresa Selva Gas Tocache SAC?</p>	<p>Específicos:</p> <p>Determinar la influencia del sistema web en el nivel de servicio para el proceso de control de inventario en la empresa Selva Gas Tocache SAC.</p>	<p>Específicos:</p> <p>El sistema web mejora el nivel de servicio en el proceso de control de inventario en la empresa Selva Gas Tocache SAC.</p>	<p>Dependiente:</p> <p>Proceso de control de inventario</p>	Dimensiones	Indicadores	Instrumento	Fórmula	<p><b>Población:</b></p> <p>I1: Nivel de servicio – 540 pedidos</p> <p>I2: Índice de rotación de Stock - 600 productos</p> <p><b>Muestra:</b></p> <p>I1: Nivel de servicio - 225 pedidos</p> <p>I2: Índice de rotación de Stock - 234 productos</p> <p><b>Método de investigación:</b></p> <p>Hipotético - Deductivo</p>
				Entrega	Nivel de servicio	Ficha de registro	$\text{Nivel de servicio (\%)} = \frac{\text{Ventas}}{\text{Demandas}} \times 100$	
				Almacén	Índice de rotación de Stock	Ficha de registro	$\text{RPT (\%)} = \frac{\text{Ventas a precio coste}}{\text{Saldo medio de productos terminados}} \times 100$ <p><b>Donde:</b></p> <p><b>RPT %:</b> Porcentaje del índice de rotación de stock</p>	
<p>¿De qué manera influye un sistema web en el índice de rotación de stock para el proceso de control de inventario en la empresa Selva Gas Tocache SAC?</p>	<p>Determinar la influencia del sistema web en el índice de rotación de stock para el proceso de control de inventario en la empresa Selva Gas Tocache SAC.</p>	<p>El sistema web mejora el índice de rotación de stock en el proceso de control de inventario en la empresa Selva Gas Tocache SAC.</p>						

### Anexo N°3: Instrumento de recolección de datos

Ficha de Registro de Actividades					
Investigadores	ROMERO RODRIGUEZ BILLY JERSON				
Empresa	SELVA GAS TOCACHE S.A.C				
Fecha	01 - noviembre - 2020 al 30 - noviembre - 2020				
Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Nivel de servicio	Determina el porcentaje del nivel de servicio de almacén	Fichaje	Razón	Ficha de registro de actividades	$\text{Nivel de servicio (\%)} = \frac{\text{Ventas}}{\text{Demandas}} \times 100$
PRE-TEST					
Item	Fecha	Proceso de evaluación		Porcentaje del nivel de servicio	
		Ventas	Demanda	Nivel de servicio (%) = $\frac{\text{Ventas}}{\text{Demandas}} \times 100$	
1	01-nov	6	8	75	
2	02-nov	6	8	75	
3	03-nov	8	9	89	
4	04-nov	7	9	78	
5	05-nov	7	10	70	
6	06-nov	8	11	73	
7	07-nov	9	12	75	
8	08-nov	7	9	78	
9	09-nov	7	10	70	
10	10-nov	8	9	89	
11	11-nov	8	10	80	
12	12-nov	10	12	83	
13	13-nov	7	8	88	
14	14-nov	8	11	73	
15	15-nov	7	10	70	
16	16-nov	8	10	80	
17	17-nov	7	10	70	
18	18-nov	6	9	67	
19	19-nov	8	10	80	
20	20-nov	7	9	78	
21	21-nov	8	10	80	
22	22-nov	10	12	83	
23	23-nov	7	9	78	
24	24-nov	8	10	80	
25	25-nov	7	10	70	
26	26-nov	7	12	58	
27	27-nov	7	10	70	
28	28-nov	6	8	75	
29	29-nov	9	11	82	
30	30-nov	7	10	70	

Ficha de Registro de Actividades					
Investigadores	ROMERO RODRIGUEZ BILLY JERSON				
Empresa	SELVA GAS TOCACHE S.A.C				
Fecha	01 - junio - 2021 al 30 - junio - 2021				
Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Nivel de servicio	Determina el porcentaje del nivel de servicio de almacén	Fichaje	Razón	Ficha de registro de actividades	$\text{Nivel de servicio (\%)} = \frac{\text{Ventas}}{\text{Demandas}} \times 100$
POST-TEST					
Item	Fecha	Proceso de evaluación		Porcentaje del nivel de servicio	
		Ventas	Demanda	Nivel de servicio (%) = $\frac{\text{Ventas}}{\text{Demandas}} \times 100$	
1	01-jun	9	10	90	
2	02-jun	8	10	80	
3	03-jun	10	11	91	
4	04-jun	9	9	100	
5	05-jun	10	11	91	
6	06-jun	8	9	89	
7	07-jun	10	11	91	
8	08-jun	9	10	90	
9	09-jun	9	11	82	
10	10-jun	10	10	100	
11	11-jun	9	10	90	
12	12-jun	11	12	92	
13	13-jun	9	9	100	
14	14-jun	8	9	89	
15	15-jun	7	8	88	
16	16-jun	9	11	82	
17	17-jun	8	10	80	
18	18-jun	8	9	89	
19	19-jun	8	9	89	
20	20-jun	9	10	90	
21	21-jun	9	10	90	
22	22-jun	11	12	92	
23	23-jun	9	10	90	
24	24-jun	8	8	100	
25	25-jun	8	8	100	
26	26-jun	9	10	90	
27	27-jun	9	9	100	
28	28-jun	8	9	89	
29	29-jun	10	11	91	
30	30-jun	9	10	90	

Ficha de Registro de Actividades					
Investigadores	ROMERO RODRIGUEZ BILLY JERSON				
Empresa	SELVA GAS TOCACHE S.A.C				
Fecha	01 - noviembre - 2020 al 30 - noviembre - 2020				
Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Índice de rotación de Stock	Determina el índice de rotación de Stock	Fichaje	Razón	Ficha de registro de actividades	$RPT = \frac{\text{Ventas a precio coste}}{\text{Saldo medio de productos terminados}}$
PRE-TEST					
Item	Fecha	Proceso de evaluación		RPT = (Ventas a precio coste) / (Saldo medio de productos terminados)	
		Ventas a precio coste	Saldo medio de productos terminados		
1	01-nov	7	26	0.27	
2	02-nov	7	19	0.37	
3	03-nov	8	57	0.14	
4	04-nov	7	49	0.14	
5	05-nov	8	42	0.19	
6	06-nov	8	34	0.24	
7	07-nov	9	26	0.35	
8	08-nov	7	57	0.12	
9	09-nov	7	50	0.14	
10	10-nov	9	43	0.21	
11	11-nov	8	34	0.24	
12	12-nov	10	26	0.38	
13	13-nov	7	56	0.13	
14	14-nov	8	49	0.16	
15	15-nov	9	41	0.22	
16	16-nov	8	32	0.25	
17	17-nov	7	24	0.29	
18	18-nov	7	52	0.13	
19	19-nov	8	45	0.18	
20	20-nov	7	37	0.19	
21	21-nov	8	30	0.27	
22	22-nov	10	22	0.45	
23	23-nov	7	47	0.15	
24	24-nov	8	40	0.20	
25	25-nov	7	32	0.22	
26	26-nov	8	25	0.32	
27	27-nov	7	17	0.41	
28	28-nov	7	40	0.18	
29	29-nov	9	33	0.27	
30	30-nov	7	24	0.29	

Ficha de Registro de Actividades					
Investigadores	ROMERO RODRIGUEZ BILLY JERSON				
Empresa	SELVA GAS TOCACHE S.A.C				
Fecha	01 - junio - 2021 al 30 - junio - 2021				
Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Índice de rotación de Stock	Determina el índice de rotación de Stock	Fichaje	Razón	Ficha de registro de actividades	$RPT = \frac{\text{Ventas a precio coste}}{\text{Saldo medio de productos terminados}}$
POST-TEST					
Item	Fecha	Proceso de evaluación		RPT = (Ventas a precio coste) / (Saldo medio de productos terminados)	
		Ventas a precio coste	Saldo medio de productos terminados		
1	01-jun	11	21	0.52	
2	02-jun	9	10	0.90	
3	03-jun	10	51	0.20	
4	04-jun	10	41	0.24	
5	05-jun	11	31	0.35	
6	06-jun	9	20	0.45	
7	07-jun	10	11	0.91	
8	08-jun	11	55	0.20	
9	09-jun	9	44	0.20	
10	10-jun	10	35	0.29	
11	11-jun	11	25	0.44	
12	12-jun	12	14	0.86	
13	13-jun	10	47	0.21	
14	14-jun	8	37	0.22	
15	15-jun	9	29	0.31	
16	16-jun	9	20	0.45	
17	17-jun	10	11	0.91	
18	18-jun	8	48	0.17	
19	19-jun	9	40	0.23	
20	20-jun	10	31	0.32	
21	21-jun	9	21	0.43	
22	22-jun	11	12	0.92	
23	23-jun	9	48	0.19	
24	24-jun	10	39	0.26	
25	25-jun	8	29	0.28	
26	26-jun	10	21	0.48	
27	27-jun	9	11	0.82	
28	28-jun	9	42	0.21	
29	29-jun	10	33	0.30	
30	30-jun	11	23	0.48	

#### Anexo N°4: Calculo del tamaño de la muestra

Para el cálculo de la muestra se utilizará según Hernández et al. (2010) la siguiente fórmula:

$$n = \frac{NZ^2 PQ}{(N - 1)e^2 + z^2 PQ}$$

Donde:

n: Muestra.

N: Población.

Z: Nivel de confianza (95%)

P: Probabilidad de éxito (50%)

Q: Probabilidad de fracaso (50%).

e: Error (5%).

Aplicando la fórmula anterior se calculan las muestras para cada indicador:

Indicador 1: Nivel de servicio

$$n = \frac{540 * 1.96^2 * 0.5 * 0.5}{(540 - 1)0.05^2 + 1.96^2 * 0.5 * 0.5}$$

n= 224.71 = 225 pedidos

Indicador 2: Índice de rotación de Stock

$$n = \frac{600 * (1.96)^2 * (0.5) * (0.5)}{(600 - 1)(0.05)^2 + (1.96)^2 * (0.5) * (0.5)}$$

n= 234.44 = 234 productos

## Anexo N°5: Validación del instrumento de investigación

### TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Villaverde Medrano, Hugo

Título y/o Grado:

PhD. ....( )	Doctor.....(x)	Magister..( )	Ingeniero ( )	Licenciado ( )	Otro. ( )
--------------	----------------	---------------	---------------	----------------	-----------

Autores: Romero Rodríguez Billy Jerson

Universidad Cesar Vallejo

Fecha: 09/05/2021

### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

#### Tabla de evaluación de expertos para el indicador: nivel de servicio

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas en la cual tendrá que poner el "%".

ITEMS	PREGUNTAS	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy Bueno 71 – 80 %	Excelente 81 – 100 %
1	¿El instrumento de medición cumple con el diseño adecuado?				75%	
2	¿El instrumento de recolección de datos tiene relación con el título de Investigación?				75%	
3	¿En el instrumento de recolección se mencionan las variables de Investigación?				75%	
4	¿El diseño del instrumento de medición facilitara el análisis y procesamiento de datos?				75%	
5	¿El instrumento de recolección de datos facilitara el logro de los objetivos de la Investigación?				75%	
6	¿El instrumento de medición es claro, preciso y sencillo para que se registre la información sin inconvenientes?				75%	
7	¿Del instrumento de medición son entendibles sus variables?				75%	
8	¿El instrumento de medición es claro, preciso y sencillo, pudiendo de esa manera obtener los datos requeridos?				75%	
9	¿El instrumento de medición será accesible a la población sujeto de estudio?				75%	
10	¿Cada una de las variables del instrumento de medición, se relacionan con cada uno de los elementos de los indicadores?				75%	
<b>TOTAL</b>					75%	

SUGERENCIAS:

\_\_\_\_\_

\_\_\_\_\_



\_\_\_\_\_  
Firma del Experto

## TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Fermín Pérez, Félix Armando

Título y/o Grado:

PhD. ....( )	Doctor.....( )	Magister..( x )	Ingeniero ( )	Licenciado ( )	Otro. ( )
--------------	----------------	-----------------	---------------	----------------	-----------

Autores: Romero Rodríguez Billy Jerson

Universidad César Vallejo

Fecha: 18/05/2021

### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

##### Tabla de evaluación de expertos para el indicador: nivel de servicio

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas en la cual tendrá que poner el "%".

ITEMS	PREGUNTAS	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy Bueno 71 – 80 %	Excelente 81 – 100 %
1	¿El instrumento de medición cumple con el diseño adecuado?					84%
2	¿El instrumento de recolección de datos tiene relación con el título de investigación?					84%
3	¿En el instrumento de recolección se mencionan las variables de investigación?					84%
4	¿El diseño del instrumento de medición facilitara el análisis y procesamiento de datos?					84%
5	¿El instrumento de recolección de datos facilitara el logro de los objetivos de la investigación?					84%
6	¿El instrumento de medición es claro, preciso y sencillo para que se registre la información sin inconvenientes?					84%
7	¿Del instrumento de medición son entendibles sus variables?					84%
8	¿El instrumento de medición es claro, preciso y sencillo, pudiendo de esa manera obtener los datos requeridos?					84%
9	¿El instrumento de medición será accesible a la población sujeto de estudio?					84%
10	¿Cada una de las variables del instrumento de medición, se relacionan con cada uno de los elementos de los indicadores?					84%
<b>TOTAL</b>						84%

SUGERENCIAS:

---



---

*Fermín Pérez*

Félix Armando Fermín Pérez



### TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Johnson Romero, Guillermo Miguel

Título y/o Grado:

PhD. .... ( )	Doctor..... ( )	Magister..(X)	Ingeniero ( )	Licenciado ( )	Otro. ( )
---------------	-----------------	---------------	---------------	----------------	-----------

Autores: Romero Rodríguez Billy Jerson

Universidad Cesar Vallejo

Fecha: 20/05/2021

#### TÍTULO DEL PROYECTO

**DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C**

Tabla de evaluación de expertos para el indicador: nivel de servicio

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas en la cual tendrá que poner el "%".

ITEMS	PREGUNTAS	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy Bueno 71 – 80 %	Excelente 81 – 100 %
1	¿El instrumento de medición cumple con el diseño adecuado?					83%
2	¿El instrumento de recolección de datos tiene relación con el título de investigación?					83%
3	¿En el instrumento de recolección se mencionan las variables de investigación?					83%
4	¿El diseño del instrumento de medición facilitara el análisis y procesamiento de datos?					83%
5	¿El instrumento de recolección de datos facilitara el logro de los objetivos de la investigación?					83%
6	¿El instrumento de medición es claro, preciso y sencillo para que se registre la información sin inconvenientes?					83%
7	¿Del instrumento de medición son entendibles sus variables?					83%
8	¿El instrumento de medición es claro, preciso y sencillo, pudiendo de esa manera obtener los datos requeridos?					83%
9	¿El instrumento de medición será accesible a la población sujeto de estudio?					83%
10	¿Cada una de las variables del instrumento de medición, se relacionan con cada uno de los elementos de los indicadores?					83%
<b>TOTAL</b>						83%

SUGERENCIAS:

---



Firma del Experto

### TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Villaverde Medrano, Hugo

Título y/o Grado:

PhD. ....( )	Doctor.....(x)	Magister..( )	Ingeniero ( )	Licenciado ( )	Otro. ( )
					_____

Autores: Romero Rodríguez Billy Jerson

Universidad Cesar Vallejo

Fecha: 09/05/2021

### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

Tabla de evaluación de expertos para el indicador: índice de rotación de stock

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas en la cual tendrá que poner el "%".

ITEMS	PREGUNTAS	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy Bueno 71 – 80 %	Excelente 81 – 100 %
1	¿El instrumento de medición cumple con el diseño adecuado?				75%	
2	¿El instrumento de recolección de datos tiene relación con el título de Investigación?				75%	
3	¿En el instrumento de recolección se mencionan las variables de Investigación?				75%	
4	¿El diseño del instrumento de medición facilitara el análisis y procesamiento de datos?				75%	
5	¿El instrumento de recolección de datos facilitara el logro de los objetivos de la Investigación?				75%	
6	¿El instrumento de medición es claro, preciso y sencillo para que se registre la información sin inconvenientes?				75%	
7	¿Del instrumento de medición son entendibles sus variables?				75%	
8	¿El instrumento de medición es claro, preciso y sencillo, pudiendo de esa manera obtener los datos requeridos?				75%	
9	¿El instrumento de medición será accesible a la población sujeto de estudio?				75%	
10	¿Cada una de las variables del instrumento de medición, se relacionan con cada uno de los elementos de los indicadores?				75%	
<b>TOTAL</b>					75%	

SUGERENCIAS:

\_\_\_\_\_



Firma del Experto

## TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Fermín Pérez, Félix Armando

Título y/o Grado:

PhD. ....( )	Doctor.....( )	Magister..( x )	Ingeniero ( )	Licenciado ( )	Otro. ( )
--------------	----------------	-----------------	---------------	----------------	-----------

Autores: Romero Rodríguez Billy Jerson

Universidad Cesar Vallejo

Fecha: 18/05/2021

### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

##### Tabla de evaluación de expertos para el indicador: índice de rotación de stock

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas en la cual tendrá que poner el "%".

ITEMS	PREGUNTAS	Deficiente	Regular	Bueno	Muy Bueno	Excelente
		0 – 20 %	21 – 50 %	51 – 70 %	71 – 80 %	81 – 100 %
1	¿El instrumento de medición cumple con el diseño adecuado?					84%
2	¿El instrumento de recolección de datos tiene relación con el título de investigación?					84%
3	¿En el instrumento de recolección se mencionan las variables de investigación?					84%
4	¿El diseño del instrumento de medición facilitara el análisis y procesamiento de datos?					84%
5	¿El instrumento de recolección de datos facilitara el logro de los objetivos de la investigación?					84%
6	¿El instrumento de medición es claro, preciso y sencillo para que se registre la información sin inconvenientes?					84%
7	¿Del instrumento de medición son entendibles sus variables?					84%
8	¿El instrumento de medición es claro, preciso y sencillo, pudiendo de esa manera obtener los datos requeridos?					84%
9	¿El instrumento de medición será accesible a la población sujeto de estudio?					84%
10	¿Cada una de las variables del instrumento de medición, se relacionan con cada uno de los elementos de los indicadores?					84%
<b>TOTAL</b>						84%

SUGERENCIAS:

---



---

*Fermín Pérez*

Félix Armando Fermín Pérez

## TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Johnson Romero, Guillermo Miguel

Título y/o Grado:

PhD. ....( )	Doctor.....( )	Magister..( X )	Ingeniero ( )	Licenciado ( )	Otro. ( )
--------------	----------------	-----------------	---------------	----------------	-----------

Autores: Romero Rodriguez Billy Jerson

Universidad Cesar Vallejo

Fecha: 20/05/2021

### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

Tabla de evaluación de expertos para el indicador: índice de rotación de stock

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar los instrumentos utilizados para medir un indicador, mediante una serie de preguntas en la cual tendrá que poner el "%".

ITEMS	PREGUNTAS	Deficiente 0 – 20 %	Regular 21 – 50 %	Bueno 51 – 70 %	Muy Bueno 71 – 80 %	Excelente 81 – 100 %
1	¿El instrumento de medición cumple con el diseño adecuado?					81%
2	¿El instrumento de recolección de datos tiene relación con el título de Investigación?					81%
3	¿En el instrumento de recolección se mencionan las variables de investigación?					81%
4	¿El diseño del instrumento de medición facilitara el análisis y procesamiento de datos?					81%
5	¿El instrumento de recolección de datos facilitara el logro de los objetivos de la investigación?					81%
6	¿El instrumento de medición es claro, preciso y sencillo para que se registre la información sin inconvenientes?					81%
7	¿Del instrumento de medición son entendibles sus variables?					81%
8	¿El instrumento de medición es claro, preciso y sencillo, pudiendo de esa manera obtener los datos requeridos?					81%
9	¿El instrumento de medición será accesible a la población sujeto de estudio?					81%
10	¿Cada una de las variables del instrumento de medición, se relacionan con cada uno de los elementos de los indicadores?					81%
<b>TOTAL</b>						81%

SUGERENCIAS:



Firma del Experto

## Anexo N°6: Resultados de la confiabilidad de los instrumentos

Ficha de Registro de Actividades					
Investigadores	ROMERO RODRIGUEZ BILLY JERSON				
Empresa	SELVA GAS TOCACHE S.A.C				
Fecha	01 - octubre - 2020 al 30 - octubre -2020				
Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Nivel de servicio	Determina el porcentaje del nivel de servicio de almacén	Fichaje	Razón	Ficha de registro de actividades	$\text{Nivel de servicio (\%)} = \frac{\text{Ventas}}{\text{Demandas}} \times 100$
TEST					
Item	Fecha	Proceso de evaluación		Porcentaje del nivel de servicio	
		Ventas	Demanda	Nivel de servicio (%) = $\frac{\text{Ventas}}{\text{Demandas}} \times 100$	
1	01-oct	6	8	75	
2	02-oct	7	9	78	
3	03-oct	9	10	90	
4	04-oct	7	9	78	
5	05-oct	9	11	82	
6	06-oct	7	10	70	
7	07-oct	6	9	67	
8	08-oct	6	8	75	
9	09-oct	7	10	70	
10	10-oct	8	9	89	
11	11-oct	9	11	82	
12	12-oct	7	9	78	
13	13-oct	6	9	67	
14	14-oct	8	11	73	
15	15-oct	7	10	70	
16	16-oct	7	9	78	
17	17-oct	6	9	67	
18	18-oct	6	9	67	
19	19-oct	9	11	82	
20	20-oct	8	10	80	
21	21-oct	8	10	80	
22	22-oct	9	11	82	
23	23-oct	7	9	78	
24	24-oct	10	12	83	
25	25-oct	6	9	67	
26	26-oct	8	13	62	
27	27-oct	7	10	70	
28	28-oct	9	11	82	
29	29-oct	10	12	83	
30	30-oct	6	9	67	

*Patricia*

Ficha de Registro de Actividades	
Investigadores	ROMERO RODRIGUEZ BILLY JERSON
Empresa	SELVA GAS TOCACHE S.A.C
Fecha	01 - noviembre - 2020 al 30 - noviembre - 2020

Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Nivel de servicio	Determina el porcentaje del nivel de servicio de almacén	Fichaje	Razón	Ficha de registro de actividades	Nivel de servicio (%) =(Ventas )/Demandas x100

RE-TEST					
Item	Fecha	Proceso de evaluación			Porcentaje del nivel de servicio Nivel de servicio (%) = (Ventas )/Demandas x100
		Ventas	Demanda		
1	01-nov	6	8		75
2	02-nov	6	8		75
3	03-nov	8	9		89
4	04-nov	7	9		78
5	05-nov	7	10		70
6	06-nov	8	11		73
7	07-nov	9	12		75
8	08-nov	7	9		78
9	09-nov	7	10		70
10	10-nov	8	9		89
11	11-nov	8	10		80
12	12-nov	10	12		83
13	13-nov	7	8		88
14	14-nov	8	11		73
15	15-nov	7	10		70
16	16-nov	8	10		80
17	17-nov	7	10		70
18	18-nov	6	9		67
19	19-nov	8	10		80
20	20-nov	7	9		78
21	21-nov	8	10		80
22	22-nov	10	12		83
23	23-nov	7	9		78
24	24-nov	8	10		80
25	25-nov	7	10		70
26	26-nov	7	12		58
27	27-nov	7	10		70
28	28-nov	6	8		75
29	29-nov	9	11		82
30	30-nov	7	10		70

*Patricia*

### Correlaciones

		NIVEL_SERVICIO_TEST	NIVEL_SERVICIO_RE_TEST
NIVEL_SERVICIO_TEST	Correlación de Pearson	1	,722**
	Sig. (bilateral)		,000
	N	30	30
NIVEL_SERVICIO_RE_TEST	Correlación de Pearson	,722**	1
	Sig. (bilateral)	,000	
	N	30	30

\*\* . La correlación es significativa en el nivel 0,01 (bilateral).

Para este indicador se ha obtenido un valor del coeficiente de Pearson distinto de cero con  $r=0.722$ , por esta razón se afirma que hay una correlación entre las variables y según Hernández et al. (2010), el resultado es Elevada.

Ficha de Registro de Actividades	
Investigadores	ROMERO RODRIGUEZ BILLY JERSON
Empresa	SELVA GAS TOCACHE S.A.C
Fecha	01 - octubre - 2020 al 30 - octubre - 2020

Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Índice de rotación de Stock	Determina el índice de rotación de Stock	Fichaje	Razón	Ficha de registro de actividades	$RPT = \frac{\text{Ventas a precio coste}}{\text{Saldo medio de productos terminados}}$

TEST				
Item	Fecha	Proceso de evaluación		RPT = $\frac{\text{Ventas a precio coste}}{\text{Saldo medio de productos terminados}}$
		Ventas a precio coste	Saldo medio de productos terminados	
1	01-oct	6	25	0.24
2	02-oct	7	19	0.37
3	03-oct	9	52	0.17
4	04-oct	7	43	0.16
5	05-oct	9	36	0.25
6	06-oct	7	27	0.26
7	07-oct	8	20	0.40
8	08-oct	8	52	0.15
9	09-oct	7	44	0.16
10	10-oct	8	37	0.22
11	11-oct	9	29	0.31
12	12-oct	7	20	0.35
13	13-oct	6	53	0.11
14	14-oct	8	47	0.17
15	15-oct	7	39	0.18
16	16-oct	7	32	0.22
17	17-oct	8	25	0.32
18	18-oct	6	52	0.12
19	19-oct	9	46	0.20
20	20-oct	8	37	0.22
21	21-oct	8	29	0.28
22	22-oct	9	21	0.43
23	23-oct	7	52	0.13
24	24-oct	10	45	0.22
25	25-oct	8	35	0.23
26	26-oct	8	27	0.30
27	27-oct	7	19	0.37
28	28-oct	9	47	0.19
29	29-oct	10	38	0.26
30	30-oct	7	28	0.25

*Patricia*

Ficha de Registro de Actividades					
Investigadores	ROMERO RODRIGUEZ BILLY JERSON				
Empresa	SELVA GAS TOCACHE S.A.C				
Fecha	01 - noviembre - 2020 al 30 - noviembre - 2020				
Indicador	Descripción	Técnica	Unidad de medida	Instrumento	Formula
Índice de rotación de Stock	Determina el índice de rotación de Stock.	Fichaje	Razón	Ficha de registro de actividades	$RPT = (Ventas \text{ a precio coste}) / (\text{Saldo medio de productos terminados})$
RE-TEST					
Item	Fecha	Proceso de evaluación		RPT = (Ventas a precio coste) / (Saldo medio de productos terminados)	
		Ventas a precio coste	Saldo medio de productos terminados		
1	01-nov	7	26		0.27
2	02-nov	7	19		0.37
3	03-nov	8	57		0.14
4	04-nov	7	49		0.14
5	05-nov	8	42		0.19
6	06-nov	8	34		0.24
7	07-nov	9	26		0.35
8	08-nov	7	57		0.12
9	09-nov	7	50		0.14
10	10-nov	9	43		0.21
11	11-nov	8	34		0.24
12	12-nov	10	26		0.38
13	13-nov	7	56		0.13
14	14-nov	8	49		0.16
15	15-nov	9	41		0.22
16	16-nov	8	32		0.25
17	17-nov	7	24		0.29
18	18-nov	7	52		0.13
19	19-nov	8	45		0.18
20	20-nov	7	37		0.19
21	21-nov	8	30		0.27
22	22-nov	10	22		0.45
23	23-nov	7	47		0.15
24	24-nov	8	40		0.20
25	25-nov	7	32		0.22
26	26-nov	8	25		0.32
27	27-nov	7	17		0.41
28	28-nov	7	40		0.18
29	29-nov	9	33		0.27
30	30-nov	7	24		0.29



### Correlaciones

		INDICE_ROTACION_STOCK_TEST	INDICE_ROTACION_STOCK_RE_TEST
INDICE_ROTACION_STOCK_TEST	Correlación de Pearson	1	,943**
	Sig. (bilateral)		,000
	N	30	30
INDICE_ROTACION_STOCK_RE_TEST	Correlación de Pearson	,943**	1
	Sig. (bilateral)	,000	
	N	30	30

\*\* . La correlación es significativa en el nivel 0,01 (bilateral).

Para este indicador se ha obtenido un valor del coeficiente de Pearson distinto de cero con  $r=0.798$ , por esta razón se afirma que hay una correlación entre las variables y según Hernández et al. (2010), el resultado es Elevada.



## Anexo N°7: Constancia de autorización para realizar el trabajo de investigación



EL GERENTE GENERAL DE LA EMPRESA SELVA GAS TOCACHE S.A.C

Estupiñan Palma, Patricia Lesly con DNI: 40483016

HACE CONSTAR:

El estudiante Romero Rodríguez Billy Jerson, viene realizando el desarrollo de la investigación de pregrado titulado: "Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C", de la escuela profesional de Ingeniería de Sistemas de la Universidad Cesar Vallejo Sede Lima Norte, en nuestra organización.

Dicha investigación será desarrollada durante el Semestre académico 2021-I, del décimo ciclo académico de la facultad de ingeniería. Se expide la presente constancia para los fines pertinentes.

Lima 5 de Abril del 2021



---


Firma

## Anexo N°8: Entrevista al gerente de la empresa Selva Gas Tocache S.A.C

Nombre del entrevistado	Patricia Lesly Estupiñan Palma
Cargo	Gerente General
Fecha	09/04/2021

- 1. ¿A qué se dedica la empresa?**  
La empresa Selva Gas Tocache S.A.C, se dedica a la venta de balones de gas de 5kg, 10kg y 45kg, al público en general.
- 2. ¿Qué procesos se realizan para realizar el control del inventario?**  
Por el momento solo lo realizamos al momento en que el cliente llega a realizar una compra, y vamos en busca del producto al almacén y verificamos si contamos o no con el producto que está buscando.
- 3. ¿Estos procesos no tienen inconvenientes?**  
Si, a veces no nos damos cuenta que falta stock de algún producto, no llevamos un registro de cada vez que se realiza alguna salida de dichos productos, y esto ocasiona que no halla como satisfacer al cliente.
- 4. ¿Cuántos productos promedio vende al día?**  
Alrededor de 20 productos al día.
- 5. ¿Cómo se realiza la venta de los productos?**  
A través de una boleta o factura, dependiendo del cliente, y se despacha el producto y cantidad que solicita, previamente se tiene que ir a buscar el producto en el almacén.
- 6. ¿Cómo verifican si el stock de cierto producto se está agotando?**  
De vez en cuando, no tenemos un horario fijo o cronograma para realizarlo.
- 7. ¿Realizan reportes cada cierto tiempo para verificar el estado de sus finanzas?**  
Solamente, al final de cada mes sacamos la cuenta para verificar las ganancias y reabastecemos nuevamente.



**8. ¿Realizan reportes para saber qué productos son los más vendidos?**

No, solo realizamos mensualmente cuando ingresos hubo y si se logró vender todo.

**9. ¿En qué consiste realizar un préstamo de un producto?**

El cliente quiere realizar una compra de un balón de gas de 10kg, pero no tiene un balón vacío, en este caso se le presta el balón hasta que nuevamente regrese a realizar la compra y así sucesivamente.

**10. ¿Cuáles son los productos que se pueden prestar a los clientes?**

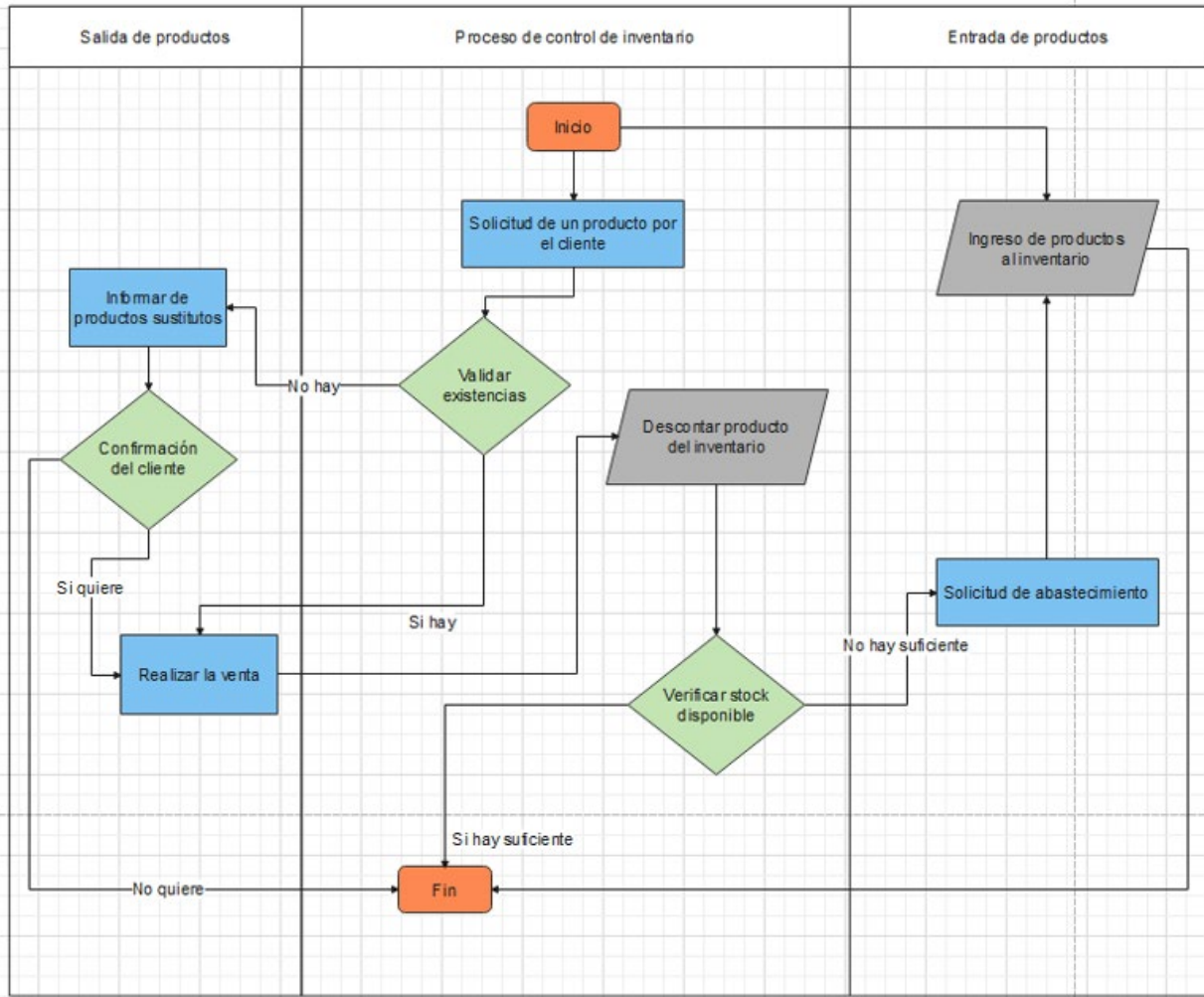
Solamente los balones de 10kg.

**11. ¿Cómo verifican si dicho cliente devolvió ese préstamo?**

Apuntamos los nombres de los clientes, pero como no llevamos un registro exacto, a veces no sabemos si el cliente regreso a devolver el balón, y no tenemos forma de saberlo ya que dichos apuntes se extravían o deterioran.

A handwritten signature in black ink, appearing to read "Patieta".

## Anexo N°9: Proceso de control de inventario



*Patricia*

## Anexo N°10: Validación de la metodología

### TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Villaverde Medrano, Hugo

Título y/o Grado:

PhD. ....( )	Doctor.....(x)	Magister..( )	Ingeniero ( )	Licenciado ( )	Otro. ( )
--------------	----------------	---------------	---------------	----------------	-----------

Universidad Cesar Vallejo

Fecha: 09/05/2021

#### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

Evaluación de la metodología para el desarrollo del sistema Web

Autor: Romero Rodríguez Billy Jerson

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de preguntas en escala del 1 a 5, siendo 1 la menor calificación y 5 la mayor calificación.

ITEMS	PREGUNTAS	METODOLOGÍA			
		SCRUM	RUP	XP	OBSERVACIONES
1	Busca desarrollar software de calidad	4	2	3	
2	Sistema flexible y de fácil adaptación	5	5	3	
3	Arroja resultados rápidos	4	4	2	
4	El cliente participa en el desarrollo	4	2	3	
5	Entrega de avances por cada etapa durante el desarrollo del software	4	4	3	
6	Entregas al cliente en corto tiempo	4	3	3	
7	Uso de menos artefactos durante la documentación del sistema	4	4	3	
8	Prioriza los requerimientos fundamentales	4	4	3	
TOTAL		33	28	23	

SUGERENCIAS:

---

---

Firma del Experto

## TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Fermín Pérez, Félix Armando

Título y/o Grado: Magister

PhD. ... ( )	Doctor.....( )	Magister..(x)	Ingeniero ( )	Licenciado ( )	Otro. ( ) _____
--------------	----------------	---------------	---------------	----------------	--------------------

Universidad César Vallejo

Fecha: 15/05/2021

### TÍTULO DEL PROYECTO

#### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

Evaluación de la metodología para el desarrollo del sistema Web

**Autor:** Romero Rodríguez Billy Jerson

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de preguntas en escala del 1 a 5, siendo 1 la menor calificación y 5 la mayor calificación.

ITEMS	PREGUNTAS	METODOLOGÍA			
		SCRUM	RUP	XP	OBSERVACIONES
1	Busca desarrollar software de calidad	5	5	4	
2	Sistema flexible y de fácil adaptación	5	4	3	
3	Arroja resultados rápidos	5	4	3	
4	El cliente participa en el desarrollo	5	4	4	
5	Entrega de avances por cada etapa durante el desarrollo del software	5	3	3	
6	Entregas al cliente en corto tiempo	5	3	4	
7	Uso de menos artefactos durante la documentación del sistema	4	3	3	
8	Prioriza los requerimientos fundamentales	4	4	3	
<b>TOTAL</b>		<b>38</b>	<b>30</b>	<b>27</b>	

SUGERENCIAS:

*Fermín Pérez*

.....  
Félix Armando Fermín Pérez

## TABLA DE EVALUACIÓN DE EXPERTOS

Apellidos y Nombre del experto: Johnson Romero, Guillermo Miguel

Título y/o Grado:

PhD. ....( )	Doctor.....( )	Magister..(X)	Ingeniero ( )	Licenciado ( )	Otro. ( )
--------------	----------------	---------------	---------------	----------------	-----------

Universidad Cesar Vallejo

Fecha: 20/05/2021

### TÍTULO DEL PROYECTO

### DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE CONTROL DE INVENTARIO EN LA EMPRESA SELVA GAS TOCACHE S.A.C

Evaluación de la metodología para el desarrollo del sistema Web

**Autor:** Romero Rodríguez Billy Jerson

Mediante la tabla de evaluación de expertos, usted tiene la facultad de calificar las metodologías involucradas, mediante una serie de preguntas en escala del 1 a 5, siendo 1 la menor calificación y 5 la mayor calificación.

ITEMS	PREGUNTAS	METODOLOGÍA			
		SCRUM	RUP	XP	OBSERVACIONES
1	Busca desarrollar software de calidad	4	5	4	
2	Sistema flexible y de fácil adaptación	5	4	3	
3	Arroja resultados rápidos	4	3	3	
4	El cliente participa en el desarrollo	5	4	4	
5	Entrega de avances por cada etapa durante el desarrollo del software	5	3	3	
6	Entregas al cliente en corto tiempo	5	3	4	
7	Uso de menos artefactos durante la documentación del sistema	4	3	3	
8	Prioriza los requerimientos fundamentales	4	4	4	
<b>TOTAL</b>		<b>36</b>	<b>29</b>	<b>28</b>	

SUGERENCIAS:

---



Firma del Experto

## Anexo N°11: Constancia de aprobación de la empresa



LA GERENTE GENERAL DE LA EMPRESA SELVA GAS TOCACHE S.A.C

Estupiñan Palma, Patricia Lesly con DNI: 40483016

CONSTA QUE:

El Sr. Romero Rodríguez, Billy Jerson con el DNI: 73231793, realizó correctamente el desarrollo e implementación del sistema web para el proceso de control de inventario, con las respectivas capacitaciones necesarias para el uso de los usuarios encargados de utilizar el sistema.

Lima, 25 Mayo de 2021

Firma



## **Anexo N°12: Desarrollo de la metodología de Scrum**

### **Introducción**

Para del desarrollo del sistema se hará uso de la metodología ágil denominada SCRUM, ya que se adapta mejor a los requerimientos que tendrá el sistema.

Cada uno de los entregables tendrá una duración de 2 a 4 semanas, estos se los conoce como Sprint.

### **Alcance**

El alcance del proyecto a realizar, debe cumplir con los siguientes objetivos:

- El sistema tendrá un usuario de tipo administrador predefinido, el cual solamente el gerente conocerá las credenciales.
- El sistema tendrá un módulo en el cual podrá asignar nuevas credenciales de acceso para los posteriores empleados.
- En el sistema se podrá realizar el CRUD (registrar, editar, eliminar) para los módulos de usuarios, clientes, proveedores, marca, productos, entrada de productos, salida de productos, y préstamos.
- El sistema mostrará mensajes de confirmación al momento de realizar algún registro o actualización de cualquier dato.
- El sistema revelará un mensaje de alerta al momento de querer eliminar algún dato.
- En el sistema se mostrará 2 gráficas de reportes, especificando las fechas que desea analizar.
- El sistema mostrará de colores diferentes el stock de los productos, el color verde cuando tenga stock más de 50%, el color naranja cuando tenga stock menos de 50%, y el color rojo cuando tenga stock menor al 25%.
- El sistema mostrara al inicio, cuando se logea, datos generales que hasta el momento tiene en la base de datos el sistema.

## Roles

- Scrum master: Encargado de que se trabaje según las guías de Scrum.
- Product Owner: Encargado de hacer las reuniones y definir las prioridades.
- Development Team: Encargado del desarrollo del producto.

## Historias

Las historias de usuarios son un elemento básico para aplicar metodologías ágiles y especialmente para poder aplicar SCRUM; describen una funcionalidad del software para el usuario, con el lenguaje que éste emplearía y tienen características como prioridad y tiempo.

Tabla N°13: *Historia N°1*

<b>Número:</b> 1	<b>Usuario:</b> Administrador
<b>Nombre :</b> Creación de la base de datos	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 4
<b>Descripción:</b> Como administrador quiero tener una base de datos en donde pueda guardar los datos necesarios para poder analizar los datos.	
<b>Observación:</b> Las tablas deben estar normalizadas y relacionadas entre sí de ser necesario.	

Tabla N°14: *Historia N°2*

<b>Número:</b> 2	<b>Usuario:</b> Administrador
<b>Nombre:</b> Creación de la plataforma web	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 4
<b>Descripción:</b> Como administrador quiero que el uso del sistema web tenga un entorno amigable y sencillo de usar.	
<b>Observación:</b> La plataforma estará dividido en 3 sectores, en la parte superior se mostrará el nombre de la empresa, del usuario y el cierre de sesión, en el izquierdo estará el menú con los módulos a usar, y en el derecho se realizarán los cambios. Al iniciar sesión, solo los administradores podrán visualizar datos y reportes generales del sistema.	

Tabla N°15: *Historia N°3*

<b>Número:</b> 3	<b>Usuario:</b> Administrador
<b>Nombre:</b> Autenticación al sistema	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador quiero autenticarme al sistema a través de un usuario y contraseña.	
<b>Observación:</b> El sistema tendrá una credencial predefinida para el administrador y luego ingresando podrá generar nuevas credenciales.	

Tabla N°16: *Historia N°4*

<b>Número:</b> 4	<b>Usuario:</b> Administrador
<b>Nombre:</b> Establecer permisos de usuario	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador no quiero que todos los usuarios tengan acceso o puedan realizar cambios a ciertos módulos.	
<b>Observación:</b> Al generar nuevas credenciales se podrá especificar qué rol tendrá, si será administrador o solo un usuario y cada uno de esos roles ya tiene predefinido a donde tendrán acceso.	

Tabla N°17: *Historia N°5*

<b>Número:</b> 5	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de los clientes	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 3
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar los clientes.	
<b>Observación:</b>	

Tabla N°18: *Historia N°6*

<b>Número:</b> 6	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de los proveedores	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 3
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar los proveedores.	
<b>Observación:</b>	

Tabla N°19: *Historia N°7*

<b>Número:</b> 7	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de las marcas	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar las marcas.	
<b>Observación:</b>	

Tabla N°20: *Historia N°8*

<b>Número:</b> 8	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de los productos	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 4
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar los productos.	
<b>Observación:</b>	

Tabla N°21: *Historia N°9*

<b>Número:</b> 9	<b>Usuario:</b> Administrador
<b>Nombre:</b> Alerta de stock mínimo de productos	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador quiero saber si está por acabarse el stock de algún producto, y no tener inconvenientes al momento de realizar una salida de producto.	
<b>Observación:</b> Cuanto el stock de algún producto tiene más del 50% se vera de color verde, si tiene menos del 50% se vera de color naranja, y rojo si tiene menos del 25%.	

Tabla N°22: *Historia N°10*

<b>Número:</b> 10	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de las entradas de productos	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 4
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar las entradas de productos.	
<b>Observación:</b>	

Tabla N°23: *Historia N°11*

<b>Número:</b> 11	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de las salidas de productos	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 5
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar las salidas de productos.	
<b>Observación:</b>	

Tabla N°24: *Historia N°12*

<b>Número:</b> 12	<b>Usuario:</b> Administrador
<b>Nombre:</b> Lista de productos de una salida de productos	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar los productos de alguna salida de productos en específico, sin tener que eliminar la salida general.	
<b>Observación:</b> La salida se generará al momento de agregarle el cliente, luego los productos que quiera pedir dicho cliente serán listados en un cuadro diferente donde se podrá registrar, ver, actualizar y eliminar nuevos productos.	

Tabla N°25: *Historia N°13*

<b>Número:</b> 13	<b>Usuario:</b> Administrador
<b>Nombre:</b> Mantenimiento de los préstamos	
<b>Prioridad:</b> Media	<b>Tiempo:</b> 3
<b>Descripción:</b> Como administrador quiero poder registrar, ver, actualizar y eliminar los préstamos.	
<b>Observación:</b>	

Tabla N°26: *Historia N°14*

<b>Número:</b> 14	<b>Usuario:</b> Administrador
<b>Nombre:</b> Gráficos de los reportes	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 5
<b>Descripción:</b> Como administrador quiero poder generar reportes y que se visualicen en unos gráficos, de un rango específicos de meses anteriormente definidas.	
<b>Observación:</b> Se generarán reportes específicos de los datos obtenidos, cualquier reporte adicional que se quiera mostrar, se le consultara con el desarrollador del sistema para realizar los cambios.	

Tabla N°27: *Historia N°15*

<b>Número:</b> 15	<b>Usuario:</b> Administrador
<b>Nombre:</b> Confirmación de cambios de los datos	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 3
<b>Descripción:</b> Como administrador quiero que se muestre algún mensaje de confirmación al momento de realizar un registro, actualización o eliminación de los datos.	
<b>Observación:</b> Se mostrará un pequeño mensaje en la parte derecha inferior de los cuadros de datos de cada uno de los módulos al momento de realizar algún cambio.	

Tabla N°28: *Historia N°16*

<b>Número:</b> 16	<b>Usuario:</b> Administrador
<b>Nombre:</b> Verificación de datos	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador quiero que se verifiquen los datos que se ingresan al momento de registrar o modificar algún cambio de los datos de cada uno de los módulos.	
<b>Observación:</b> Al momento de generar un registro o actualización de los datos se verificará si los datos son correctos, respecto al tipo de dato que pida el campo, y si resulta ser incorrecto al momento de realizar la acción se mostrará un mensaje de error y no realizara dicha acción.	

Tabla N°29: *Historia N°17*

<b>Número:</b> 17	<b>Usuario:</b> Administrador
<b>Nombre:</b> Verificación de stock	
<b>Prioridad:</b> Alta	<b>Tiempo:</b> 2
<b>Descripción:</b> Como administrador quiero que no me acepte alguna salida de productos, si el stock de dicho producto no es suficiente.	
<b>Observación:</b> Cuando se realice una salida de productos y valla a seleccionar el producto, debajo de él se mostrará el stock actual de ese producto, y si por algún motivo se equivoca e ingresa una cantidad mayor al stock, la acción se ejecutará pero mostrara una alerta de que la cantidad ingresada es incorrecta.	

## Product Backlog

Es el conjunto de tareas que se realizarán para desarrollar del proyecto y esta ordenado en base a la prioridad del requerimiento.

Tabla N°30: *Product Backlog*

Requerimiento Funcional	N° Historia	Historia	Prioridad	Tiempo
<b>RF1:</b> Acceso a la información.	1	Creación de la base de datos	Alta	4
<b>RF2:</b> La interfaz del sistema tiene que ser intuitivo y de fácil uso.	2	Creación de la plataforma web	Alta	4
<b>RF3:</b> La interfaz estará dividida en 3 sectores, una para los módulos, otra para mostrar la información de los módulos y otra que será el cierre de sesión.				
<b>RF4:</b> El acceso al sistema se debe realizar con credenciales.	3	Autenticación al sistema	Alta	2
<b>RF5:</b> Las credenciales se ingresarán a la interfaz de login del sistema.				
<b>RF6:</b> Cada rol de usuario ya debe estar predeterminado, para agilizar la creación de nuevas credenciales.	4	Establecer permisos de usuarios	Alta	2
<b>RF7:</b> Muestre de forma rápida si un producto cuenta con suficiente stock.	9	Alerta de stock mínimo	Alta	2
<b>RF8:</b> Cada rango de cantidad será mostrado de un color diferente, para su fácil interpretación.				
<b>RF9:</b> Muestre gráficas de fácil comprensión.	14	Gráficos de los reportes	Alta	5
<b>RF10:</b> Muestre un mensaje de confirmación sobre el cambio de dato.	15	Confirmación de cambio de datos	Alta	3
<b>RF11:</b> No permita el ingreso de datos incorrectos en los diferentes campos de texto a rellenar.	16	Verificación de datos	Alta	2
<b>RF12:</b> Muestre el stock del producto seleccionado para su salida.	17	Verificación del stock	Alta	2

<b>RF13:</b> Permita registrar, editar y eliminar registros de clientes.	5	Mantenimiento de los clientes	Media	3
<b>RF14:</b> Permita exportar los datos.				
<b>RF15:</b> Permita registrar, editar y eliminar registros de proveedores.	6	Mantenimiento de los proveedores	Media	3
<b>RF16:</b> Permita exportar los datos.				
<b>RF17:</b> Permita registrar, editar y eliminar registros de marcas.	7	Mantenimiento de las marcas	Media	2
<b>RF18:</b> Permita exportar los datos.				
<b>RF19:</b> Permita registrar, editar y eliminar registros de productos.	8	Mantenimiento de los productos	Media	4
<b>RF20:</b> Permita exportar los datos.				
<b>RF21:</b> Permita registrar, editar y eliminar registros de las entradas de los productos.	10	Mantenimiento de las entradas de productos	Media	4
<b>RF22:</b> Permita exportar los datos.				
<b>RF23:</b> Permita registrar, editar y eliminar registros de las salidas de los productos.	11	Mantenimiento de las salidas de productos	Media	5
<b>RF24:</b> Permita exportar los datos.				
<b>RF25:</b> Permita registrar, editar y eliminar registros de la lista de una salida de productos.	12	Lista de productos de una salida de productos	Media	3
<b>RF26:</b> Permita agregar otros productos.				
<b>RF27:</b> Permita exportar los datos.				
<b>RF28:</b> Permita registrar, editar y eliminar registros de préstamos.	13	Mantenimiento de los préstamos	Media	3
<b>RF29:</b> Muestre si se realizó la devolución.				
<b>RF30:</b> Permita exportar los datos.				

## Sprint Backlog

Es la unión de los elementos mencionados en el Product Backlog y divididos para cada Sprint.

Tabla N°31: *Product Backlog*

<b>Sprint</b>	<b>Historias</b>	<b>Tiempo</b>
Sprint 1	H1: Creación de la base de datos	10
	H2: Creación de la plataforma web	
	H3: Autenticación al sistema	
Sprint 2	H4: Establecer permisos de usuarios	10
	H5: Mantenimiento de los clientes	
	H6: Mantenimiento de los proveedores	
	H7: Mantenimiento de las marcas	
	H8: Mantenimiento de los productos	



Sprint 3	H9: Alerta de stock mínimo	10
	H10: Mantenimiento de las entradas de productos	
Sprint 4	H11: Mantenimiento de las salidas de productos	11
	H12: Lista de productos de una salida de productos	
	H13: Mantenimiento de los préstamos	
Sprint 5	H14: Gráficos de los reportes	12
	H15: Confirmación de cambio de datos	
	H16: Verificación de datos	
	H17: Verificación de stock	

## Plan de trabajo

Tabla N°32: *Plan de trabajo*

Nombre de la tarea	Duración	Comienzo	Fin
<b>Proyecto</b>	<b>53 días</b>	<b>01/04/2021</b>	<b>23/05/2021</b>
<b>Sprint 1</b>	10 días	01/04/2021	10/04/2021
<b>Historia N°1: Creación de la base de datos</b>	4 días	01/04/2021	04/04/2021
Elaboración del diagrama lógico y físico.	1 día	01/04/2021	01/04/2021
Creación de las tablas mediante migración.	1 día	02/04/2021	02/04/2021
Generar las llaves foráneas de las tablas.	1 día	03/04/2021	03/04/2021
Generación del diccionario de datos	1 día	04/04/2021	04/04/2021
<b>Historia N°2: Creación de la plataforma</b>	4 días	05/04/2021	08/04/2021
Diseño y maquetación de la interfaz del login.	1 día	05/04/2021	05/04/2021
Diseño y maquetación interfaz principal del sistema.	1 día	06/04/2021	06/04/2021
Diseño y maquetación de las interfaces de cada módulo.	2 días	07/04/2021	08/04/2021
<b>Historia N°3: Autenticación al sistema</b>	2 días	09/04/2021	10/04/2021
Validación de los campos de la tabla usuarios con la BD.	1 día	09/04/2021	09/04/2021
Integración del código a la interfaz del login.	1 día	10/04/2021	10/04/2021
<b>Sprint 2</b>	10 días	11/04/2021	20/04/2021
<b>Historia N°4: Establecer permisos de usuario</b>	2 días	11/04/2021	12/04/2021
Creación de los permisos para cada rol.	1 día	11/04/2021	11/04/2021
Codificación predefinidos por el sistema.	1 día	12/04/2021	12/04/2021
<b>Historia N°5: Mantenimiento de los clientes</b>	3 días	13/04/2021	15/04/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	13/04/2021	13/04/2021

Integración del código a la interfaz del módulo.	1 día	14/04/2021	14/04/2021
Comprobación de los campos de texto para el CRUD.	1 día	15/04/2021	15/04/2021
<b>Historia N°6: Mantenimiento de los proveedores</b>	3 días	16/04/2021	18/04/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	16/04/2021	16/04/2021
Integración del código a la interfaz del módulo.	1 día	17/04/2021	17/04/2021
Comprobación de los campos de texto para el CRUD.	1 día	18/04/2021	18/04/2021
<b>Historia N°7: Mantenimiento de las marcas</b>	2 días	19/04/2021	20/04/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	19/04/2021	19/04/2021
Integración del código y comprobación de los campos con la interfaz	1 día	20/04/2021	20/04/2021
<b>Sprint 3</b>	10 días	21/04/2021	30/04/2021
<b>Historia N°8: Mantenimiento de los productos</b>	4 días	21/04/2021	24/04/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	21/04/2021	21/04/2021
Integración del código a la interfaz del módulo.	2 días	22/04/2021	23/04/2021
Comprobación de los campos de texto para el CRUD.	1 día	24/04/2021	24/04/2021
<b>Historia N°9: Alerta de stock mínimo</b>	2 días	25/04/2021	26/04/2021
Elaboración del procedimiento para obtener los datos.	1 día	25/04/2021	25/04/2021
Integración del código con la interfaz.	1 día	26/04/2021	26/04/2021
<b>Historia N°10: Mantenimiento de las entradas de productos</b>	4 días	27/04/2021	30/04/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	27/04/2021	27/04/2021
Integración del código a la interfaz del módulo.	2 días	28/04/2021	29/04/2021
Comprobación de los campos de texto para el CRUD.	1 día	30/04/2021	30/04/2021
<b>Sprint 4</b>	11 días	01/05/2021	11/05/2021
<b>Historia N°11: Mantenimiento de las salidas de productos</b>	5 días	01/05/2021	05/05/2021
Elaboración de procedimientos para el CRUD de datos.	2 días	01/05/2021	02/05/2021
Integración del código a la interfaz del módulo.	2 días	03/05/2021	04/05/2021
Comprobación de los campos de texto para el CRUD.	1 día	05/05/2021	05/05/2021
<b>Historia N°12: Lista de productos de una salida de productos</b>	3 días	06/05/2021	08/05/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	06/05/2021	06/05/2021
Integración del código a la interfaz del módulo.	1 día	07/05/2021	07/05/2021
Comprobación de los campos de texto para el CRUD.	1 día	08/05/2021	08/05/2021
<b>Historia N°13: Mantenimiento de los préstamos</b>	3 días	09/05/2021	11/05/2021
Elaboración de procedimientos para el CRUD de datos.	1 día	09/05/2021	09/05/2021
Integración del código a la interfaz del módulo.	1 día	10/05/2021	10/05/2021
Comprobación de los campos de texto para el CRUD.	1 día	11/05/2021	11/05/2021
<b>Sprint 5</b>	12 días	12/05/2021	23/05/2021

<b>Historia N°14: Gráfico de los reportes</b>	5 días	12/05/2021	16/05/2021
Elaboración de procedimientos para procesar el rango de fechas	2 días	12/05/2021	13/05/2021
Comprobación de la existencia de datos.	2 días	14/05/2021	15/05/2021
Integración del código a la interfaz del módulo.	1 día	16/05/2021	16/05/2021
<b>Historia N°15: Confirmación de cambio de datos</b>	3 días	17/05/2021	19/05/2021
Confirmación del procesamiento de los datos.	2 días	17/05/2021	18/05/2021
Integración del código con la interfaz.	1 día	19/05/2021	19/05/2021
<b>Historia N°16: Verificación de datos</b>	2 días	20/05/2021	21/05/2021
Comprobación del tipo de dato según el campo.	1 día	20/05/2021	20/05/2021
Integración a la interfaz mediante un mensaje.	1 día	21/05/2021	21/05/2021
<b>Historia N°17: Verificación de stock</b>	2 días	22/05/2021	23/05/2021
Comprobación del stock con la BD.	1 día	22/05/2021	22/05/2021
Integración de la validación de stock a la interfaz.	1 día	23/05/2021	23/05/2021

## Desarrollo de Sprint

### Sprint 1

A continuación, se mostrarán los requerimientos funcionales que se realizarán para el sprint 1.

Tabla N°33: *Sprint 1*

#	Requerimientos Funcionales	N° Historia	Nombre Historia	Prioridad	Tiempo
1	Acceso a la información.	H1	Creación de la base de datos	Alta	4
2	La interfaz del sistema tiene que ser intuitivo y de fácil uso.	H2	Creación de la plataforma web	Alta	4
	La interfaz estará dividida en 3 sectores, una para los módulos, otra para mostrar la información de los módulos y otra que será el cierre de sesión.				
3	El acceso al sistema se debe realizar con credenciales.	H3	Autenticación del sistema	Alta	2
	Las credenciales se ingresarán a la interfaz de login del sistema.				
Total					10

### Planificación del Sprint 1

Siendo las 9:00 horas del día 01 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Lesly Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura de los requerimientos del sistema.

Analizada la lista de requerimientos por el señor Romero Rodríguez Billy Jerson solventa algunas dudas sobre los requerimientos y se compromete a cumplir con los requisitos establecidos por el Sprint 1.

Los presentes imparten su aprobación a los requerimientos de acuerdo a lo establecido en la planificación del Sprint 1 indicando que la fecha de entrega de este Sprint sería el día 10 de abril de 2021.



Firma

## Cronograma de tareas del Sprint 1

Tabla N°34: Cronograma del Sprint 1

<b>Sprint 1</b>	<b>Requerimiento F.</b>	10 días	01/04/2021	10/04/2021
<b>Historia N°1: Creación de la base de datos</b>	<b>RF1</b>	4 días	01/04/2021	04/04/2021
Elaboración del diagrama lógico y físico.		1 día	01/04/2021	01/04/2021
Creación de las tablas mediante migración.		1 día	02/04/2021	02/04/2021
Generar las llaves foráneas de las tablas.		1 día	03/04/2021	03/04/2021
Generación del diccionario de datos		1 día	04/04/2021	04/04/2021
<b>Historia N°2: Creación de la plataforma</b>		<b>RF2, RF3</b>	4 días	05/04/2021
Diseño y maquetación de la interfaz del Login.	1 día		05/04/2021	05/04/2021
Diseño y maquetación interfaz principal del sistema.	1 día		06/04/2021	06/04/2021
Diseño y maquetación de las interfaces de cada módulo.	2 días		07/04/2021	08/04/2021
<b>Historia N°3: Autenticación al sistema</b>	<b>RF4, RF5</b>	2 días	09/04/2021	10/04/2021
Validación de los campos de la tabla usuarios con la BD.		1 día	09/04/2021	09/04/2021
Integración del código a la interfaz del login.		1 día	10/04/2021	10/04/2021

En la tabla anterior se muestra el cronograma del Sprint 1, en donde se muestra las fechas en las que se realizarán los requerimientos funcionales con respecto a cada una de sus historias de usuario.

# Historia N° 1: Creación de la base de datos

Elaboración del diagrama lógico y físico.

Figura N°8. Diagrama lógico

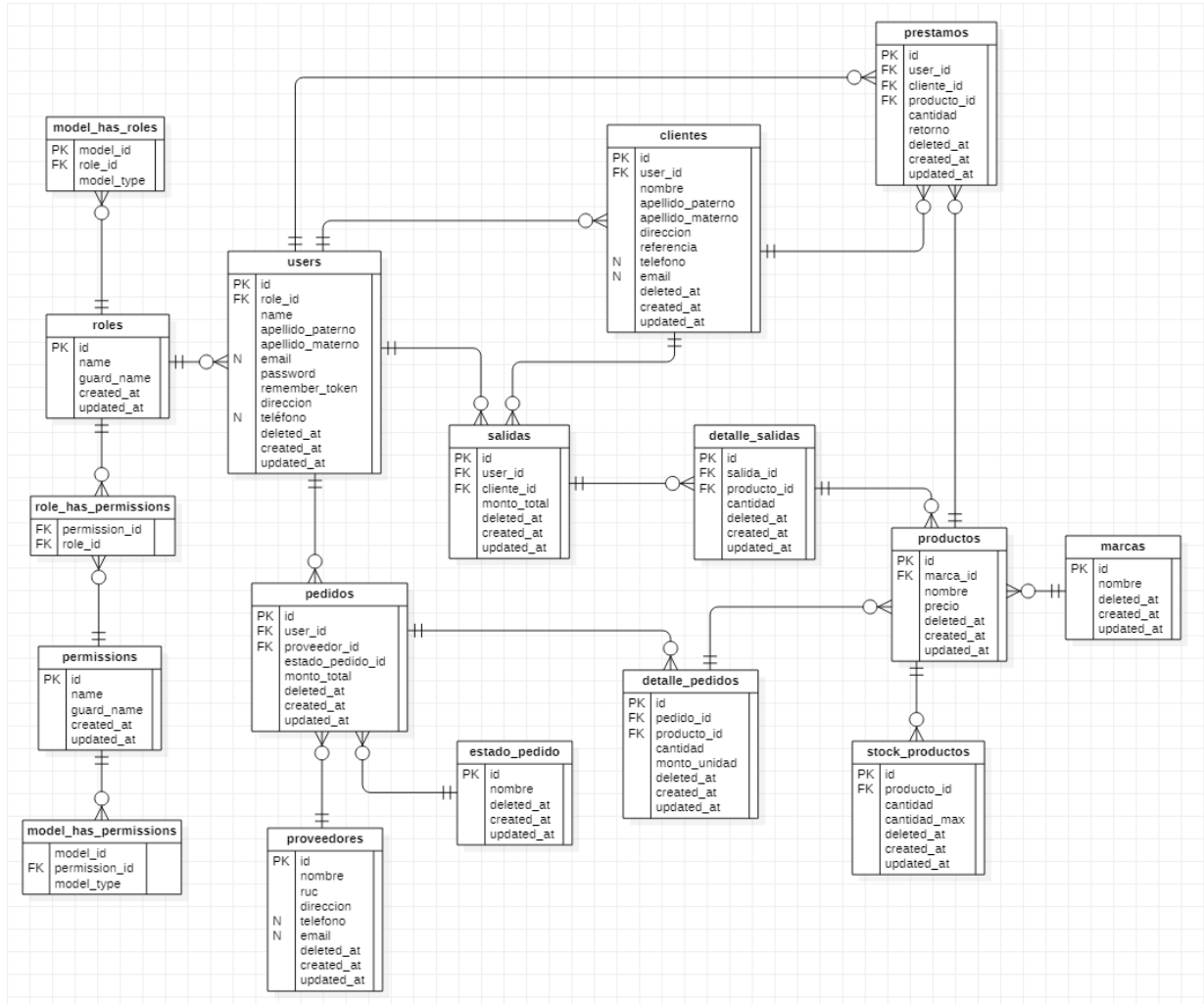
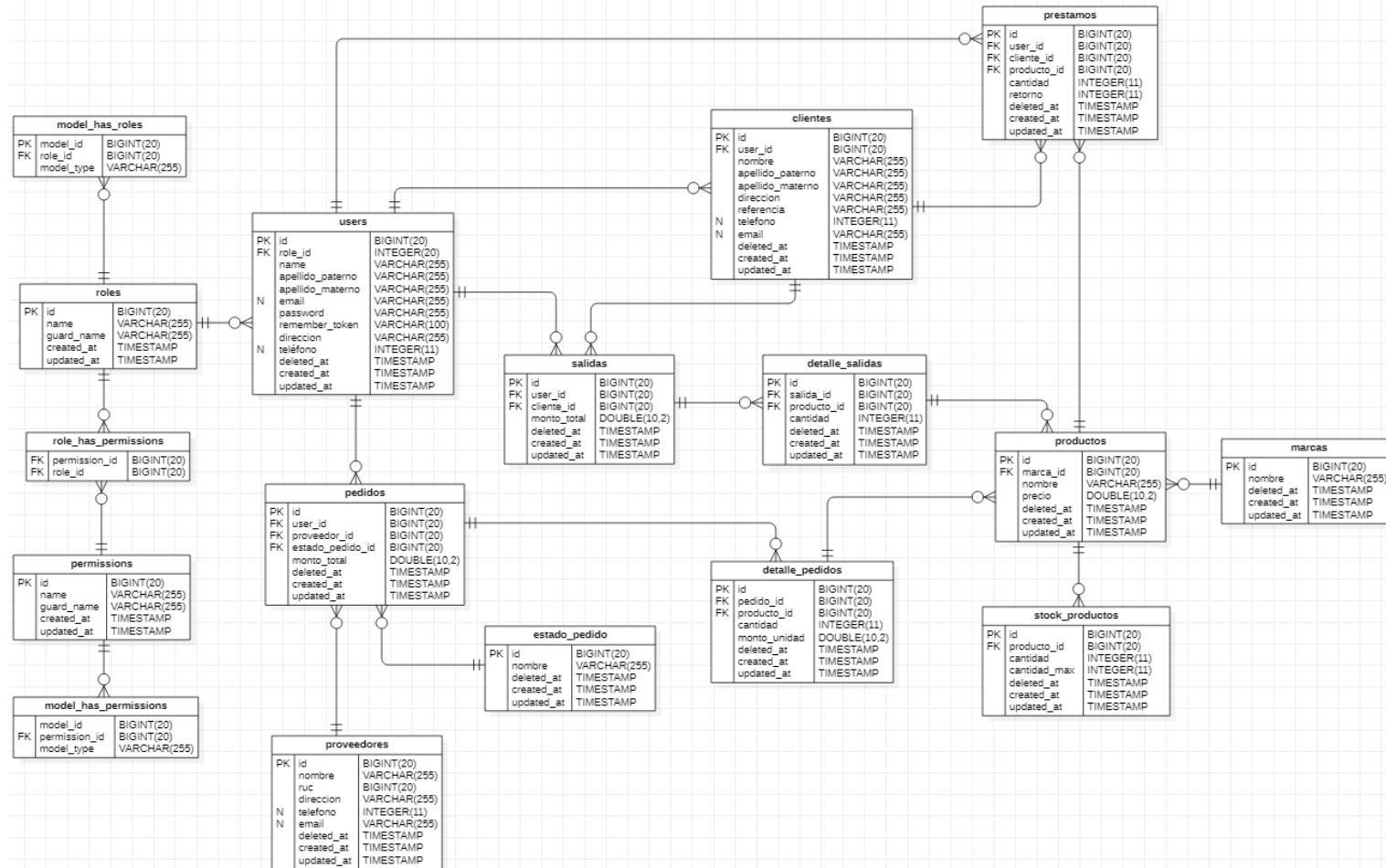


Figura N°9. Diagrama físico



Creación de las tablas mediante migración.

Figura N°10. Script de la tabla users

```
16 Schema::create('users', function (Blueprint $table) {
17     $table->engine = 'InnoDB';
18     $table->bigIncrements('id');
19     $table->string('name');
20     $table->string('apellido_paterno');
21     $table->string('apellido_materno');
22     $table->string('email')->nullable();
23     $table->string('password');
24     $table->string('direccion');
25     $table->integer('telefono')->nullable();
26     $table->rememberToken();
27     $table->softDeletes();
28     $table->timestamps();
29 });
```

Figura N°11. Script de la tabla clientes

```
17 Schema::create('clientes', function (Blueprint $table) {
18     $table->engine = 'InnoDB';
19     $table->bigIncrements('id');
20     $table->string('nombre');
21     $table->string('apellido_paterno');
22     $table->string('apellido_materno');
23     $table->string('direccion');
24     $table->string('referencia')->nullable();
25     $table->integer('telefono')->nullable();
26     $table->string('email')->nullable();
27     $table->timestamps();
28     $table->softDeletes();
29 });
```

Figura N°12. Script de la tabla prestamos

```
31 ▼ Schema::create('prestamos', function (Blueprint $table) {
32     $table->engine = 'InnoDB';
33     $table->bigIncrements('id');
34     $table->integer('cantidad');
35     $table->integer('retorno')->default(0);
36     $table->timestamps();
37     $table->softDeletes();
38 });
```



Figura N°13. Script de la tabla proveedores

```
40 Schema::create('proveedores', function (Blueprint $table) {
41     $table->engine = 'InnoDB';
42     $table->bigIncrements('id');
43     $table->string('nombre');
44     $table->bigInteger('ruc');
45     $table->string('direccion');
46     $table->integer('telefono')->nullable();
47     $table->string('email')->nullable();
48     $table->timestamps();
49     $table->softDeletes();
50 });
```

Figura N°14. Script de la tabla pedidos

```
52 Schema::create('pedidos', function (Blueprint $table) {
53     $table->engine = 'InnoDB';
54     $table->bigIncrements('id');
55     $table->double('monto_total',10,2);
56     $table->timestamps();
57     $table->softDeletes();
58 });
```

Figura N°15. Script de la tabla detalle\_pedidos

```
60 Schema::create('detalle_pedidos', function (Blueprint $table) {
61     $table->engine = 'InnoDB';
62     $table->bigIncrements('id');
63     $table->integer('cantidad');
64     $table->double('monto_unidad',10,2);
65     $table->timestamps();
66     $table->softDeletes();
67 });
```

Figura N°16. Script de la tabla salidas

```
69 Schema::create('salidas', function (Blueprint $table) {
70     $table->engine = 'InnoDB';
71     $table->bigIncrements('id');
72     $table->double('monto_total',10,2)->nullable();
73     $table->timestamps();
74     $table->softDeletes();
75 });
```

Figura N°17. Script de la tabla detalle\_salidas

```
77 Schema::create('detalle_salidas', function (Blueprint $table) {
78     $table->engine = 'InnoDB';
79     $table->bigIncrements('id');
80     $table->integer('cantidad');
81     $table->timestamps();
82     $table->softDeletes();
83 });
```

Figura N°18. Script de la tabla productos

```
85 Schema::create('productos', function (Blueprint $table) {
86     $table->engine = 'InnoDB';
87     $table->bigIncrements('id');
88     $table->string('nombre');
89     $table->double('precio',10,2);
90     $table->timestamps();
91     $table->softDeletes();
92 });
```

Figura N°19. Script de la tabla stock\_productos

```
94 Schema::create('stock_productos', function (Blueprint $table) {
95     $table->engine = 'InnoDB';
96     $table->bigIncrements('id');
97     $table->integer('cantidad');
98     $table->integer('cantidad_max');
99     $table->timestamps();
100    $table->softDeletes();
101 });
```

Figura N°20. Script de la tabla marcas

```
103 Schema::create('marcas', function (Blueprint $table) {
104     $table->engine = 'InnoDB';
105     $table->bigIncrements('id');
106     $table->string('nombre');
107     $table->timestamps();
108     $table->softDeletes();
109 });
```

Figura N°21. Script de la tabla estado\_pedido

```
111 Schema::create('estado_pedido', function (Blueprint $table) {
112     $table->engine = 'InnoDB';
113     $table->bigIncrements('id');
114     $table->string('nombre');
115     $table->timestamps();
116     $table->softDeletes();
117 });
```

Figura N°22. Script de la tabla permissions

```
23 Schema::create($tableNames['permissions'], function (Blueprint $table) {
24     $table->bigIncrements('id');
25     $table->string('name');
26     $table->string('guard_name');
27     $table->timestamps();
28
29     $table->unique(['name', 'guard_name']);
30 });
```

Figura N°23. Script de la tabla roles

```
32 Schema::create($tableNames['roles'], function (Blueprint $table) {
33     $table->bigIncrements('id');
34     $table->string('name');
35     $table->string('guard_name');
36     $table->timestamps();
37
38     $table->unique(['name', 'guard_name']);
39 });
```

Figura N°24. Script de la tabla model\_has\_permissions

```
41 Schema::create($tableNames['model_has_permissions'], function (Blueprint $table) use ($tableNames, $columnNames) {
42     $table->unsignedBigInteger('permission_id');
43
44     $table->string('model_type');
45     $table->unsignedBigInteger($columnNames['model_morph_key']);
46     $table->index([$columnNames['model_morph_key'], 'model_type'], 'model_has_permissions_model_id_model_type_index');
47
48     $table->foreign('permission_id')
49         ->references('id')
50         ->on($tableNames['permissions'])
51         ->onDelete('cascade');
52
53     $table->primary(['permission_id', $columnNames['model_morph_key'], 'model_type'],
54         'model_has_permissions_permission_model_type_primary');
55 });
```

Figura N°25. Script de la tabla model\_has\_roles

```
57 Schema::create($tableNames['model_has_roles'], function (Blueprint $table) use ($tableNames, $columnNames) {
58     $table->unsignedBigInteger('role_id');
59
60     $table->string('model_type');
61     $table->unsignedBigInteger($columnNames['model_morph_key']);
62     $table->index([$columnNames['model_morph_key'], 'model_type'], 'model_has_roles_model_id_model_type_index');
63
64     $table->foreign('role_id')
65         ->references('id')
66         ->on($tableNames['roles'])
67         ->onDelete('cascade');
68
69     $table->primary(['role_id', $columnNames['model_morph_key'], 'model_type'],
70         'model_has_roles_role_model_type_primary');
71 });
```

Figura N°26. Script de la tabla role\_has\_permissions

```
73 Schema::create($tableNames['role_has_permissions'], function (Blueprint $table) use ($tableNames) {
74     $table->unsignedBigInteger('permission_id');
75     $table->unsignedBigInteger('role_id');
76
77     $table->foreign('permission_id')
78         ->references('id')
79         ->on($tableNames['permissions'])
80         ->onDelete('cascade');
81
82     $table->foreign('role_id')
83         ->references('id')
84         ->on($tableNames['roles'])
85         ->onDelete('cascade');
86
87     $table->primary(['permission_id', 'role_id'], 'role_has_permissions_permission_id_role_id_primary');
88 });
```

Generar las llaves foráneas de las tablas.

Figura N°27. Script de llaves foráneas de la tabla clientes y prestamos

```
17 Schema::table('clientes', function (Blueprint $table) {
18     $table->biginteger('user_id')->unsigned()->after('id');
19     $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
20 });
21
22 Schema::table('prestamos', function (Blueprint $table) {
23     $table->biginteger('user_id')->unsigned()->after('id');
24     $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
25     $table->biginteger('cliente_id')->unsigned()->after('user_id');
26     $table->foreign('cliente_id')->references('id')->on('clientes')->onDelete('cascade');
27     $table->biginteger('producto_id')->unsigned()->after('cliente_id');
28     $table->foreign('producto_id')->references('id')->on('productos')->onDelete('cascade');
29 });
```

Figura N°28. Script de llaves foráneas de la tabla pedidos y detalle\_pedidos

```
31 Schema::table('pedidos', function (Blueprint $table) {
32     $table->biginteger('user_id')->unsigned()->after('id');
33     $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
34     $table->biginteger('proveedor_id')->unsigned()->after('user_id');
35     $table->foreign('proveedor_id')->references('id')->on('proveedores')->onDelete('cascade');
36     $table->biginteger('estado_pedido_id')->unsigned()->after('proveedor_id');
37     $table->foreign('estado_pedido_id')->references('id')->on('estado_pedido')->onDelete('cascade');
38 });
39
40 Schema::table('detalle_pedidos', function (Blueprint $table) {
41     $table->biginteger('pedido_id')->unsigned()->after('id');
42     $table->foreign('pedido_id')->references('id')->on('pedidos')->onDelete('cascade');
43     $table->biginteger('producto_id')->unsigned()->after('pedido_id');
44     $table->foreign('producto_id')->references('id')->on('productos')->onDelete('cascade');
45 });
```

Figura N°29. Script de llaves foráneas de la tabla salidas y detalle\_salidas

```
45 Schema::table('salidas', function (Blueprint $table) {
46     $table->biginteger('user_id')->unsigned()->after('id');
47     $table->foreign('user_id')->references('id')->on('users')->onDelete('cascade');
48     $table->biginteger('cliente_id')->unsigned()->after('user_id');
49     $table->foreign('cliente_id')->references('id')->on('clientes')->onDelete('cascade');
50 });
51
52 Schema::table('detalle_salidas', function (Blueprint $table) {
53     $table->biginteger('salida_id')->unsigned()->after('id');
54     $table->foreign('salida_id')->references('id')->on('salidas')->onDelete('cascade');
55     $table->biginteger('producto_id')->unsigned()->after('salida_id');
56     $table->foreign('producto_id')->references('id')->on('productos')->onDelete('cascade');
57 });
```

Figura N°30. Script de llaves foráneas de la tabla productos y stock\_productos

```

59 Schema::table('productos', function (Blueprint $table) {
60     $table->bigInteger('marca_id')->unsigned()->after('id');
61     $table->foreign('marca_id')->references('id')->on('marcas')->onDelete('cascade');
62 });
63
64 Schema::table('stock_productos', function (Blueprint $table) {
65     $table->bigInteger('producto_id')->unsigned()->after('id');
66     $table->foreign('producto_id')->references('id')->on('productos')->onDelete('cascade');
67 });

```

## Generación del diccionario de datos

Users: Tabla donde se guardará la información y las credenciales de los usuarios para que tengan acceso al sistema.

Figura N°31. Tabla users

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 role_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/>	3 name	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	4 apellido_paterno	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	5 apellido_materno	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	6 email	varchar(255)	utf8mb4_unicode_ci		Sí	NULL		
<input type="checkbox"/>	7 password	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	8 direccion	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	9 telefono	int(11)			Sí	NULL		
<input type="checkbox"/>	10 remember_token	varchar(100)	utf8mb4_unicode_ci		Sí	NULL		
<input type="checkbox"/>	11 deleted_at	timestamp			Sí	NULL		
<input type="checkbox"/>	12 created_at	timestamp			Sí	NULL		
<input type="checkbox"/>	13 updated_at	timestamp			Sí	NULL		

Cientes: Tabla donde se guardará la información de los clientes.

Figura N°32. Tabla clientes

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 id 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 user_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/>	3 nombre	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	4 apellido_paterno	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	5 apellido_materno	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	6 direccion	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	7 referencia	varchar(255)	utf8mb4_unicode_ci		Sí	NULL		
<input type="checkbox"/>	8 telefono	int(11)			Sí	NULL		
<input type="checkbox"/>	9 email	varchar(255)	utf8mb4_unicode_ci		Sí	NULL		
<input type="checkbox"/>	10 created_at	timestamp			Sí	NULL		
<input type="checkbox"/>	11 updated_at	timestamp			Sí	NULL		
<input type="checkbox"/>	12 deleted_at	timestamp			Sí	NULL		

Proveedores: Tabla donde se guardará la información de los proveedores.

Figura N°33. Tabla proveedores

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 <b>nombre</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	3 <b>ruc</b>	bigint(20)			No	Ninguna		
<input type="checkbox"/>	4 <b>direccion</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	5 <b>telefono</b>	int(11)			Sí	NULL		
<input type="checkbox"/>	6 <b>email</b>	varchar(255)	utf8mb4_unicode_ci		Sí	NULL		
<input type="checkbox"/>	7 <b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/>	8 <b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/>	9 <b>deleted_at</b>	timestamp			Sí	NULL		

Marcas: Tabla donde se guardará la información de las marcas.

Figura N°34. Tabla marcas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 <b>nombre</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	3 <b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/>	4 <b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/>	5 <b>deleted_at</b>	timestamp			Sí	NULL		

Productos: Tabla donde se guardará la información de los productos.

Figura N°35. Tabla productos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/>	1 <b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/>	2 <b>marca_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/>	3 <b>nombre</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/>	4 <b>precio</b>	double(10,2)			No	Ninguna		
<input type="checkbox"/>	5 <b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/>	6 <b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/>	7 <b>deleted_at</b>	timestamp			Sí	NULL		

Stock Productos: Tabla donde se guardará la información sobre el stock que existe de cada producto.

Figura N°36. Tabla stock\_productos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	producto_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 3	cantidad	int(11)			No	Ninguna		
<input type="checkbox"/> 4	cantidad_max	int(11)			No	Ninguna		
<input type="checkbox"/> 5	created_at	timestamp			Sí	NULL		
<input type="checkbox"/> 6	updated_at	timestamp			Sí	NULL		
<input type="checkbox"/> 7	deleted_at	timestamp			Sí	NULL		

Pedidos: Tabla donde se guardará la información de la elección del proveedor para la entrega de productos.

Figura N°37. Tabla pedidos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	user_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 3	proveedor_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 4	estado_pedido_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 5	monto_total	double(10,2)			No	Ninguna		
<input type="checkbox"/> 6	created_at	timestamp			Sí	NULL		
<input type="checkbox"/> 7	updated_at	timestamp			Sí	NULL		
<input type="checkbox"/> 8	deleted_at	timestamp			Sí	NULL		

Detalle Pedidos: Tabla donde se guardará la información detallada de la entrega de productos.

Figura N°38. Tabla detalle\_pedidos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	id 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	pedido_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 3	producto_id 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 4	cantidad	int(11)			No	Ninguna		
<input type="checkbox"/> 5	monto_unidad	double(10,2)			No	Ninguna		
<input type="checkbox"/> 6	created_at	timestamp			Sí	NULL		
<input type="checkbox"/> 7	updated_at	timestamp			Sí	NULL		
<input type="checkbox"/> 8	deleted_at	timestamp			Sí	NULL		

Salidas: Tabla donde se guardará la información de la cantidad total que gasto un cliente determinado.

Figura N°39. Tabla salidas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>user_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 3	<b>cliente_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 4	<b>monto_total</b>	double(10,2)			Sí	NULL		
<input type="checkbox"/> 5	<b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 6	<b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 7	<b>deleted_at</b>	timestamp			Sí	NULL		

Detalle Salidas: Tabla donde se guardará la información detallada de los productos solicitados por un cliente determinado.

Figura N°40. Tabla detalle\_salidas

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>salida_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 3	<b>producto_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 4	<b>cantidad</b>	int(11)			No	Ninguna		
<input type="checkbox"/> 5	<b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 6	<b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 7	<b>deleted_at</b>	timestamp			Sí	NULL		

Estado Pedido: Tabla donde se guardará la información sobre los estados en que puede estar un pedido determinado.

Figura N°41. Tabla estado\_pedido

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>nombre</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 3	<b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 4	<b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 5	<b>deleted_at</b>	timestamp			Sí	NULL		



Prestamos: Tabla donde se guardará la información del producto préstamo a un cliente determinado y si realizó la devolución correspondiente.

Figura N°42. Tabla prestamos

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>user_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 3	<b>cliente_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 4	<b>producto_id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 5	<b>cantidad</b>	int(11)			No	Ninguna		
<input type="checkbox"/> 6	<b>retorno</b>	int(11)			No	0		
<input type="checkbox"/> 7	<b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 8	<b>updated_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 9	<b>deleted_at</b>	timestamp			Sí	NULL		

Roles: Tabla donde se guardará la información de los roles.

Figura N°43. Tabla roles

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>name</b> 🔑	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 3	<b>guard_name</b> 🔑	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 4	<b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 5	<b>updated_at</b>	timestamp			Sí	NULL		

Permissions: Tabla donde se guardará la información sobre los permisos.

Figura N°44. Tabla permissions

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>id</b> 🔑	bigint(20)		UNSIGNED	No	Ninguna		AUTO_INCREMENT
<input type="checkbox"/> 2	<b>name</b> 🔑	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 3	<b>guard_name</b> 🔑	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 4	<b>created_at</b>	timestamp			Sí	NULL		
<input type="checkbox"/> 5	<b>updated_at</b>	timestamp			Sí	NULL		

Role has permissions: Tabla donde se guardará la información sobre los permisos de cada rol.

Figura N°45. Tabla *role\_has\_permissions*

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>permission_id</b>	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 2	<b>role_id</b>	bigint(20)		UNSIGNED	No	Ninguna		

Model has roles: Tabla donde se guardará la información sobre los modelos de cada rol.

Figura N°46. Tabla *model\_has\_roles*

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>role_id</b>	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 2	<b>model_type</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 3	<b>model_id</b>	bigint(20)		UNSIGNED	No	Ninguna		

Model has permissions: Tabla donde se guardará la información sobre los permisos de cada modelo.

Figura N°47. Tabla *model\_has\_permissions*

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra
<input type="checkbox"/> 1	<b>permission_id</b>	bigint(20)		UNSIGNED	No	Ninguna		
<input type="checkbox"/> 2	<b>model_type</b>	varchar(255)	utf8mb4_unicode_ci		No	Ninguna		
<input type="checkbox"/> 3	<b>model_id</b>	bigint(20)		UNSIGNED	No	Ninguna		

## Historia N°2: Creación de la plataforma

Diseño y maquetación de la interfaz del Login.

Figura N°48. Prototipo Login

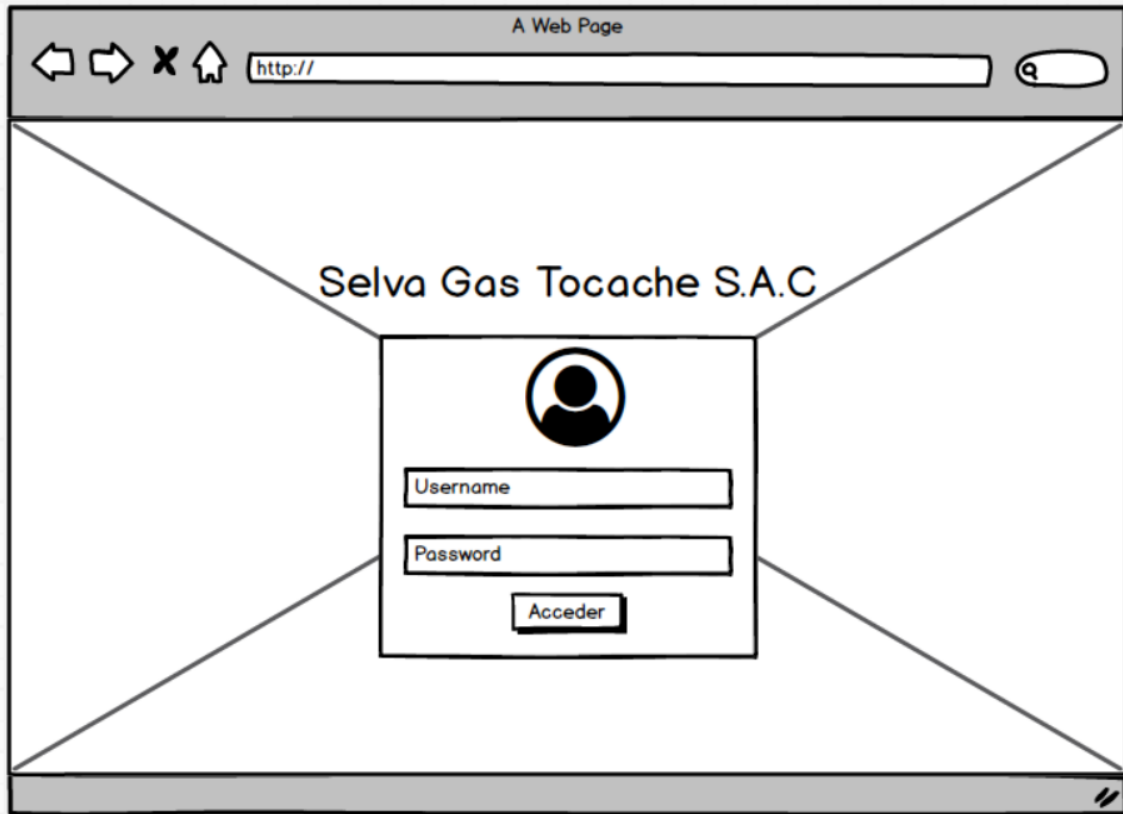
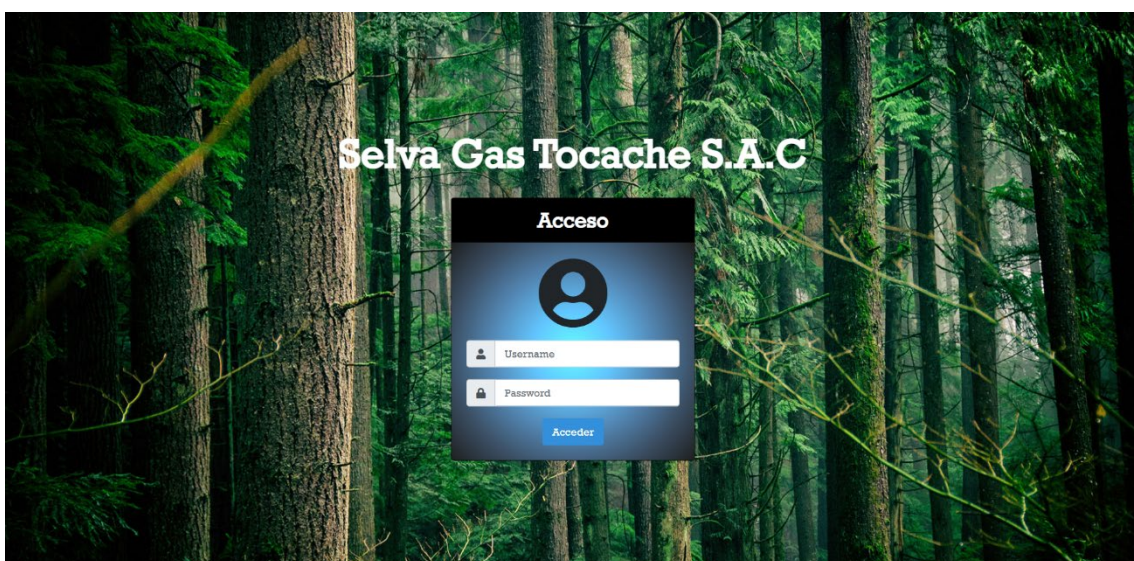


Figura N°49. Maquetación de la interfaz del Login



Diseño y maquetación interfaz principal del sistema.

Figura N°50. Prototipo principal del sistema

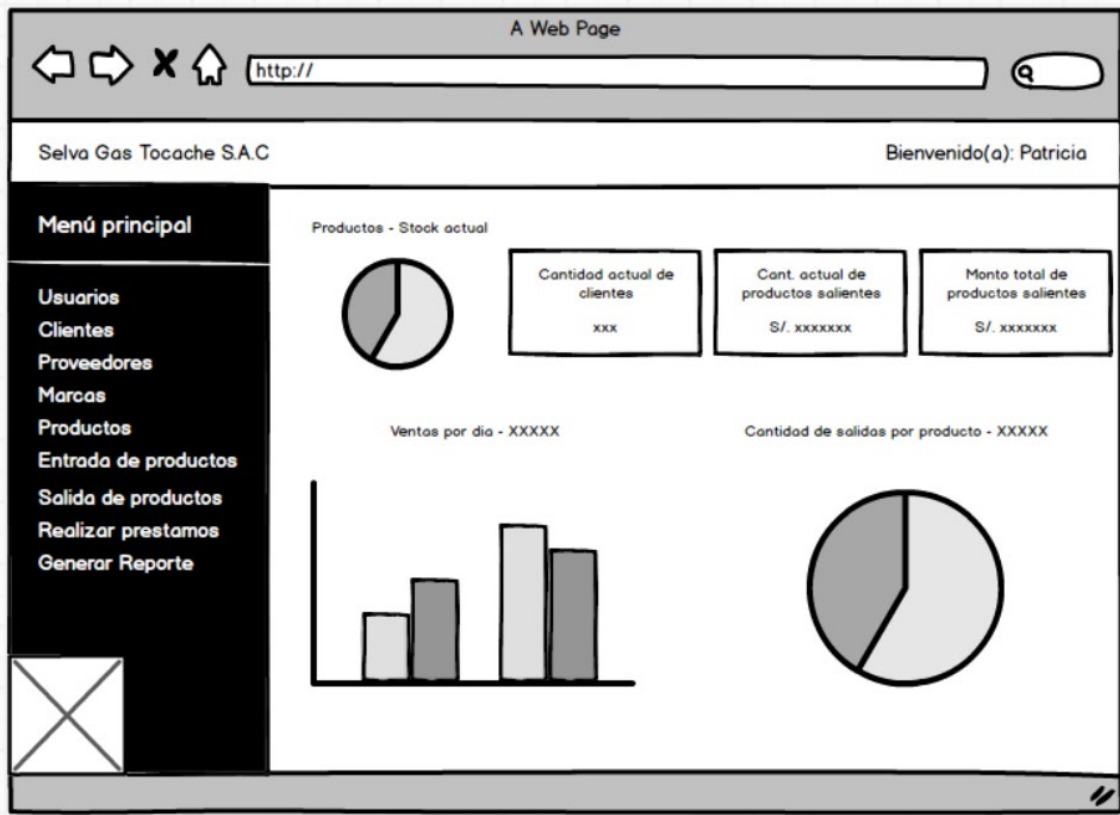


Figura N°51. Maquetación de la interfaz principal del sistema



Diseño y maquetación de las interfaces de cada módulo.

Figura N°52. Prototipo del módulo usuarios

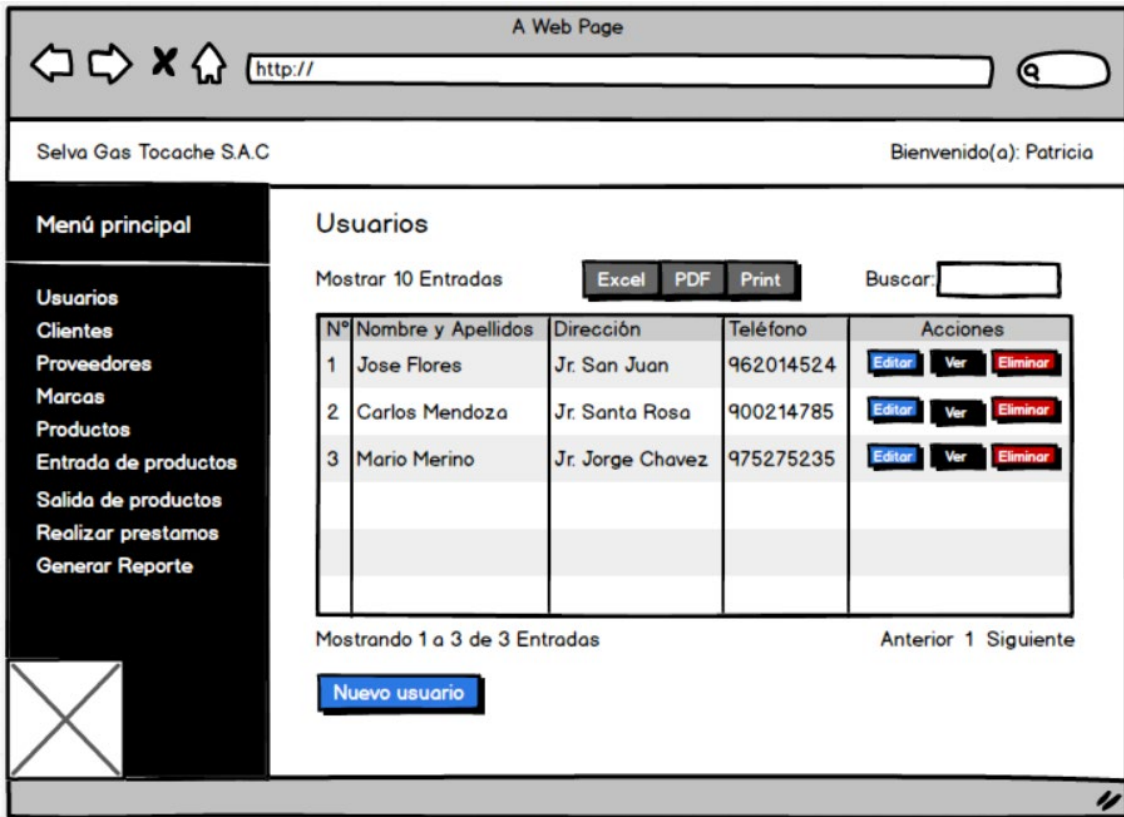


Figura N°53. Maquetación del módulo usuarios

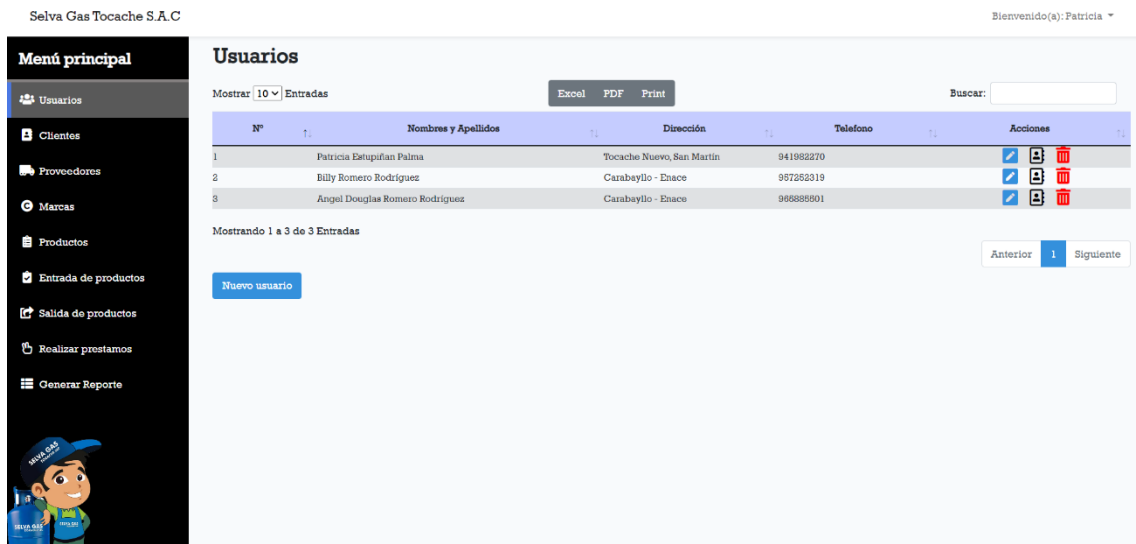


Figura N°54. Prototipo del módulo usuarios - registrar

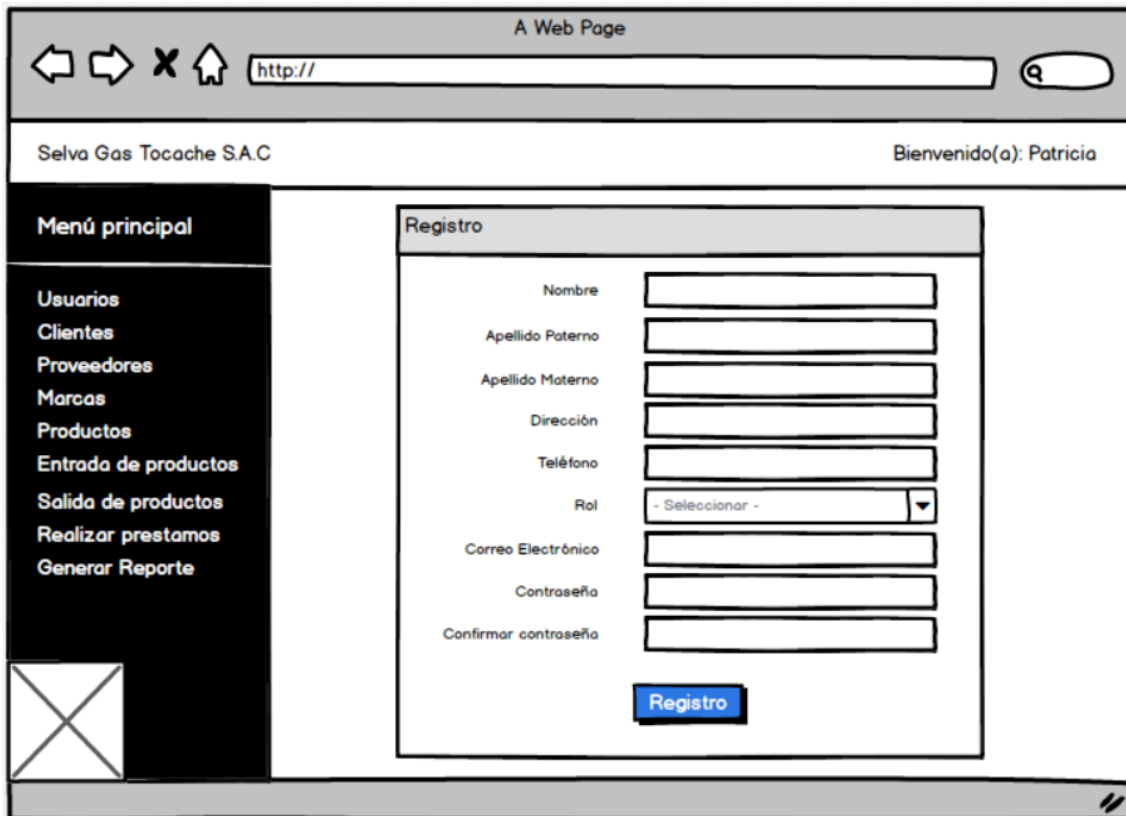


Figura N°55. Maquetación del módulo usuarios - registrar

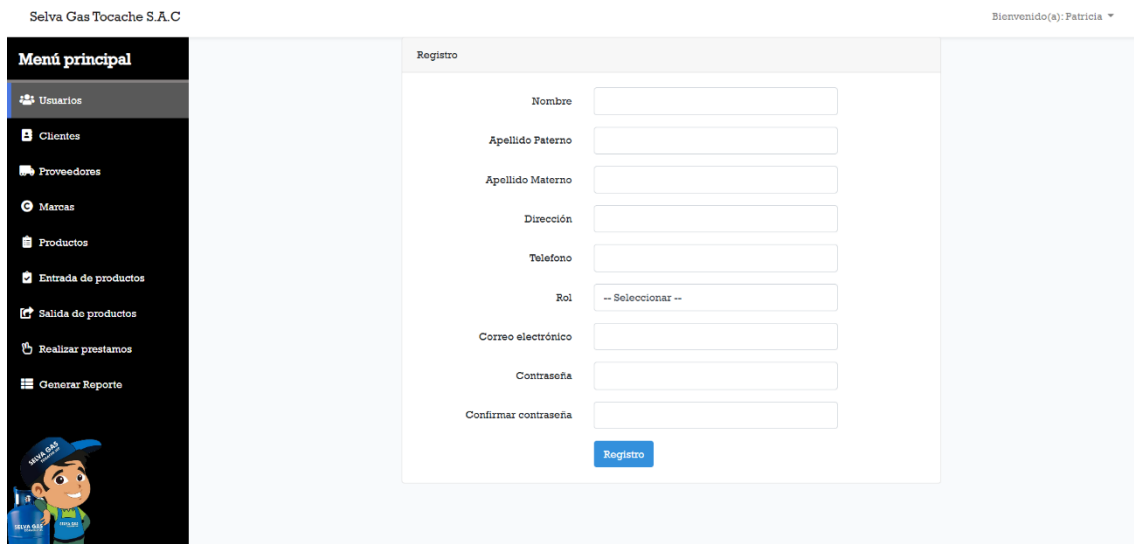


Figura N°56. Prototipo del módulo usuarios - editar

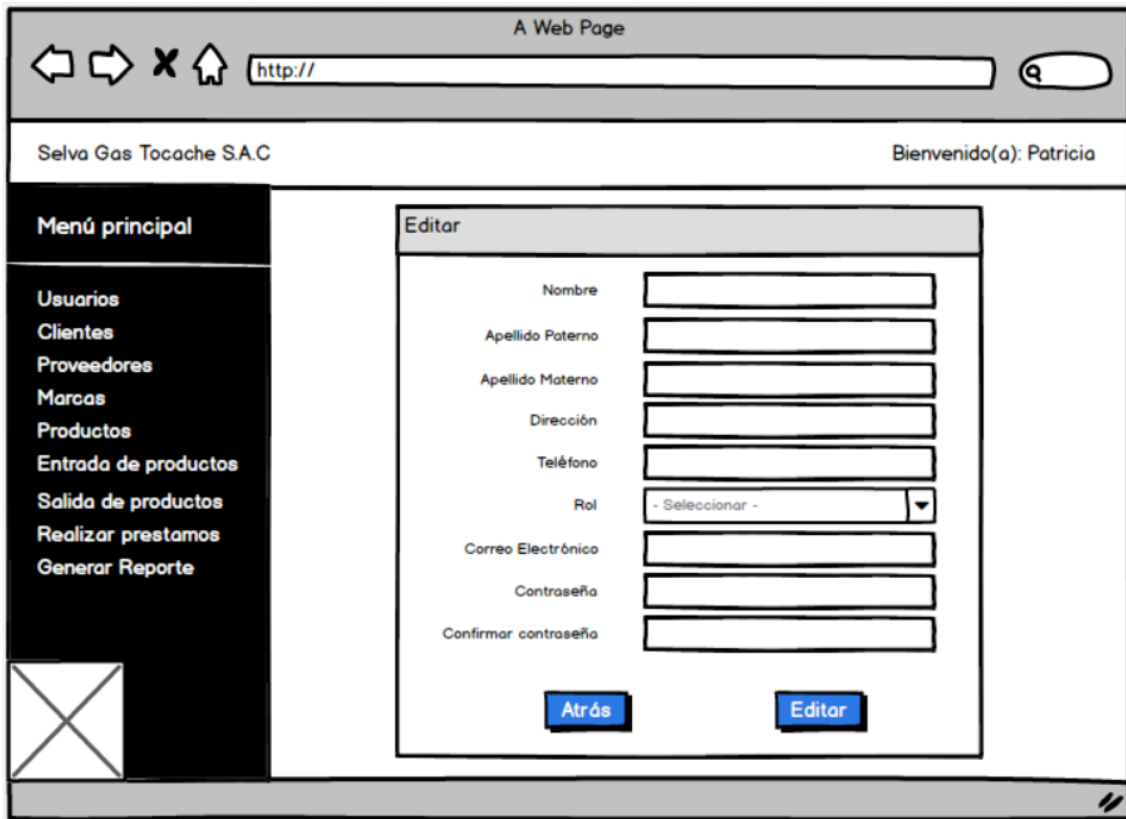


Figura N°57. Maquetación del módulo usuarios - editar

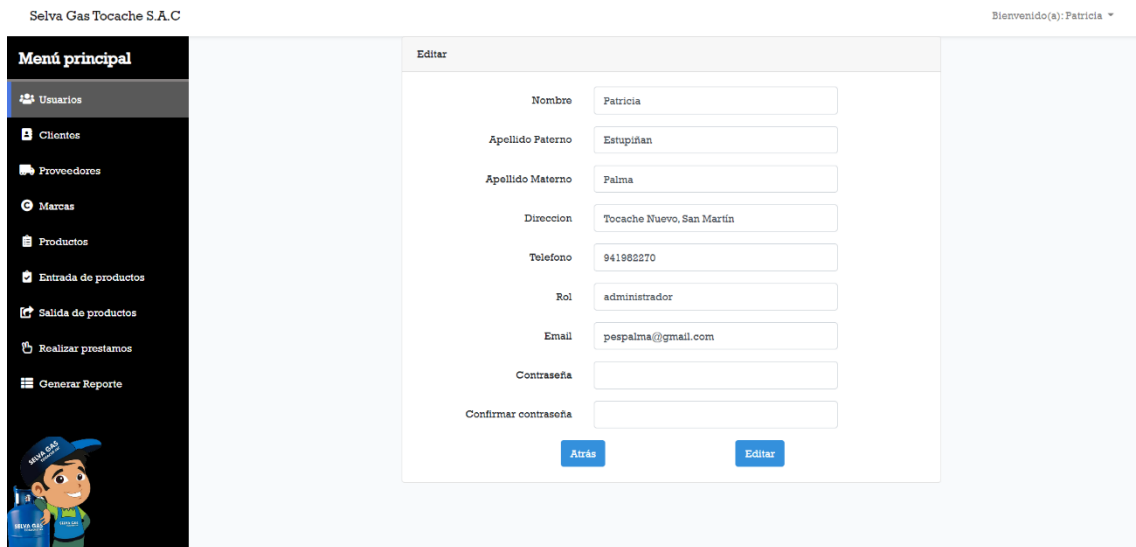


Figura N°58. Prototipo del módulo usuarios - eliminar

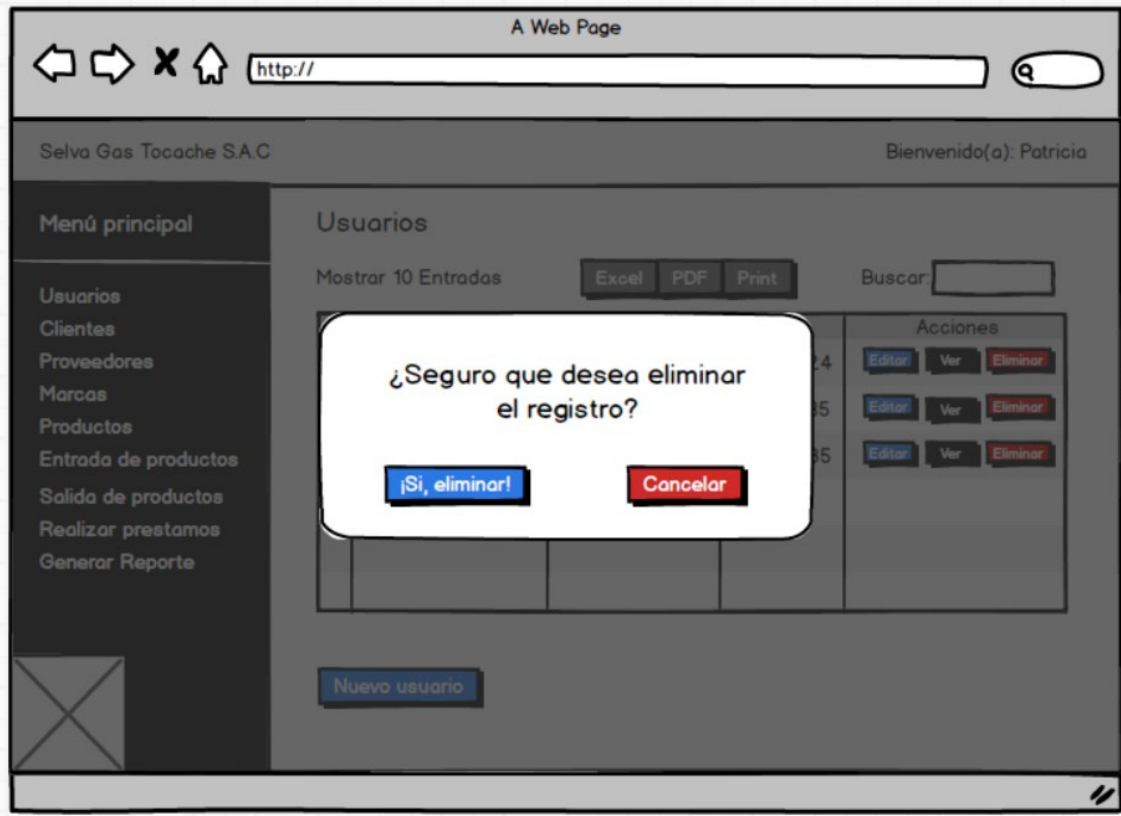


Figura N°59. Maquetación del módulo usuarios - eliminar

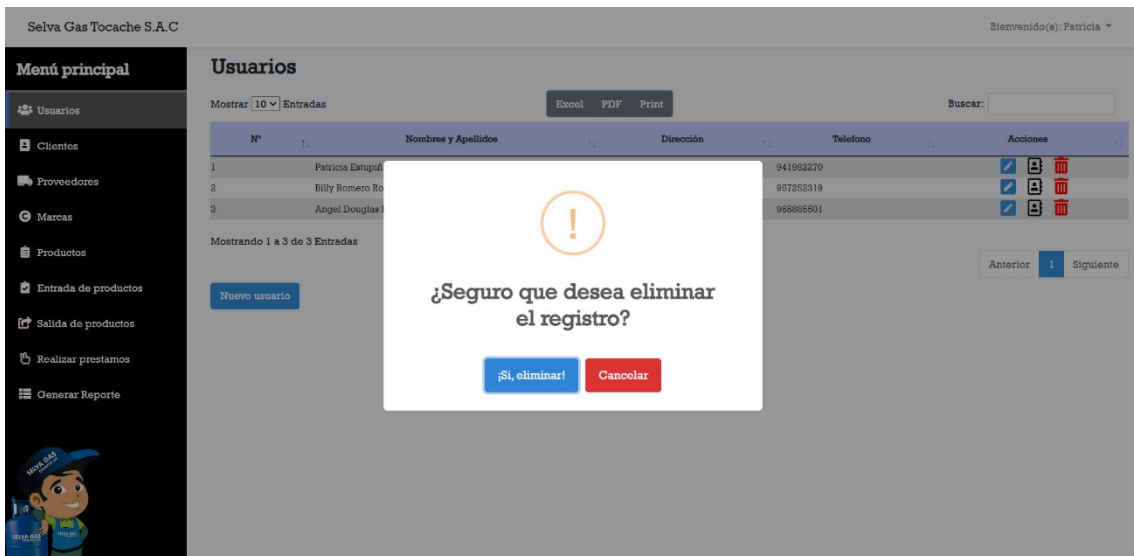




Figura N°60. Prototipo del módulo cliente

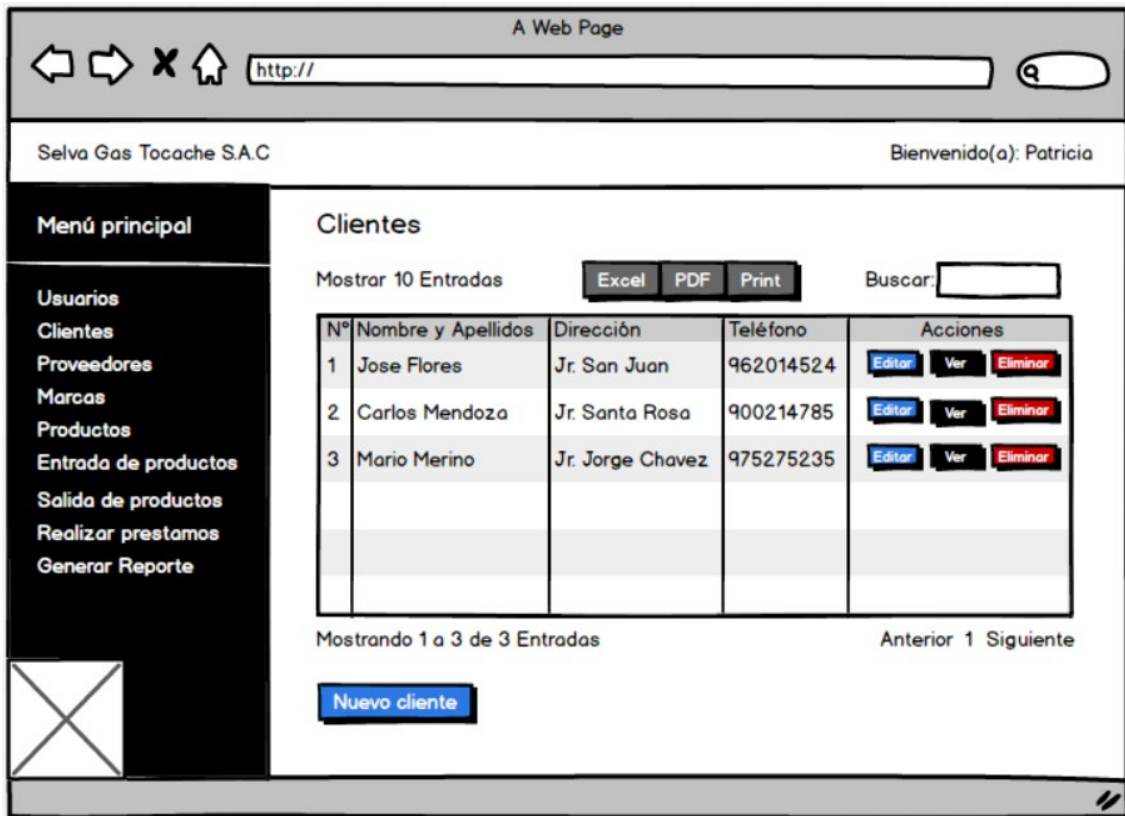


Figura N°61. Maquetación del módulo cliente

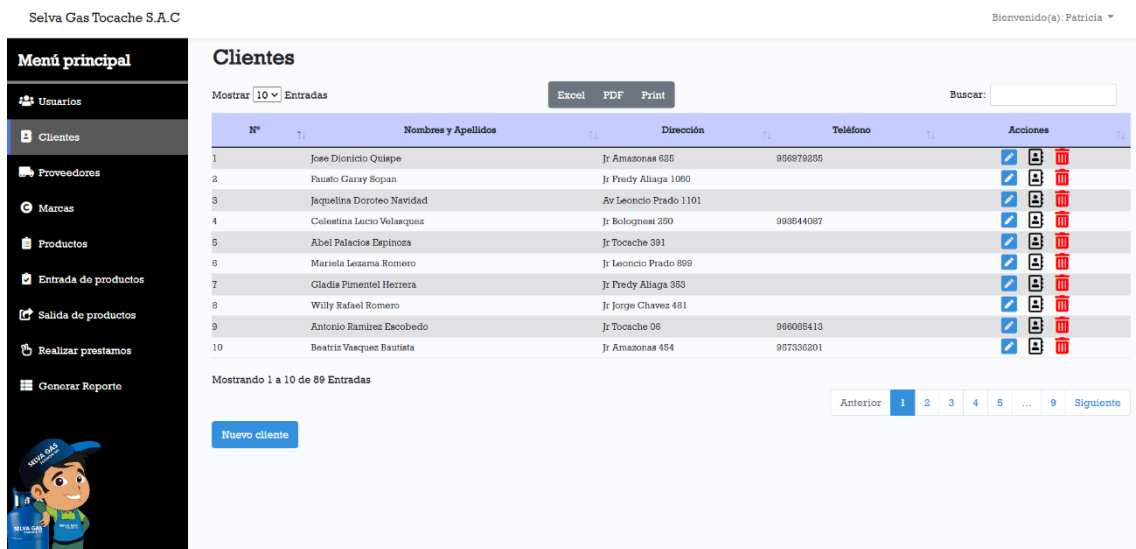


Figura N°62. Prototipo del módulo cliente - registrar

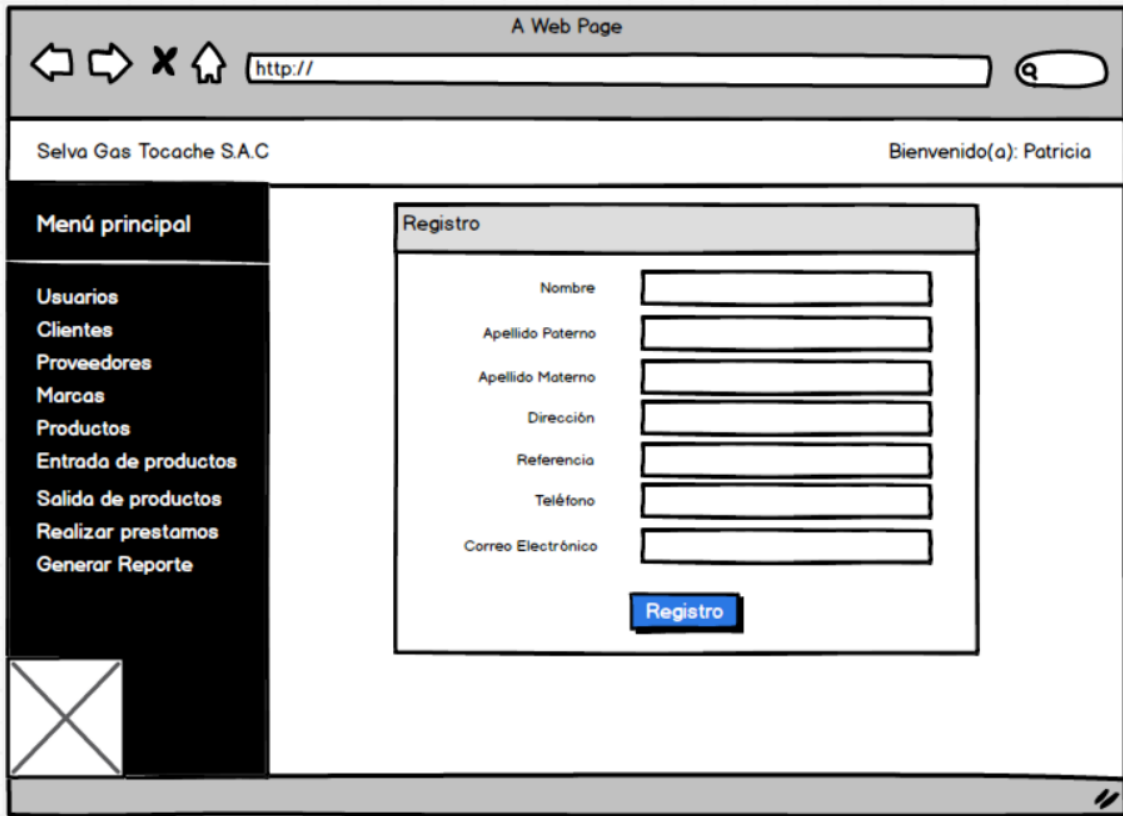


Figura N°63. Maquetación del módulo cliente - registrar

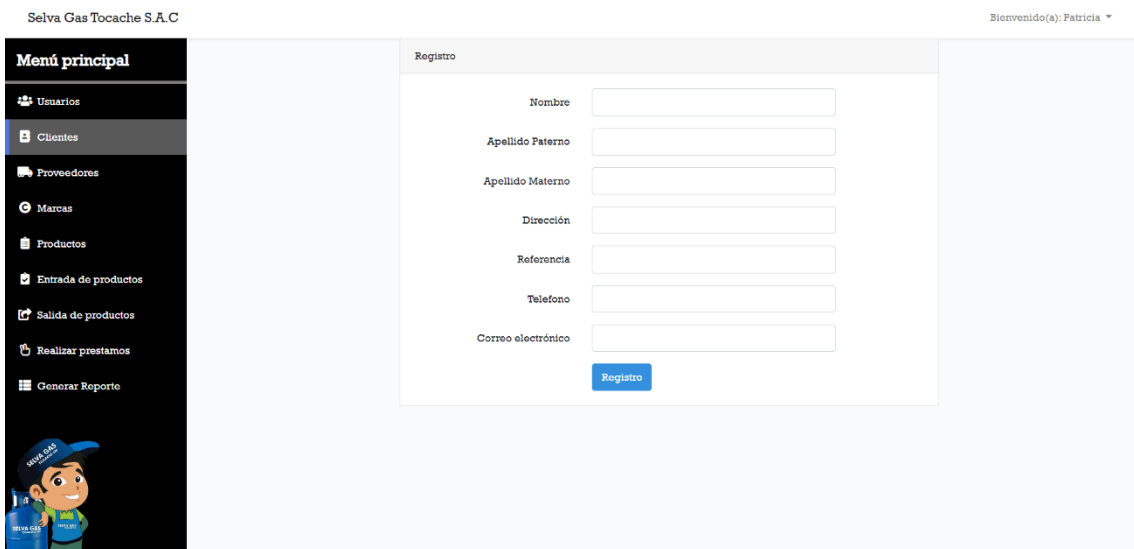


Figura N°64. Prototipo del módulo cliente - editar

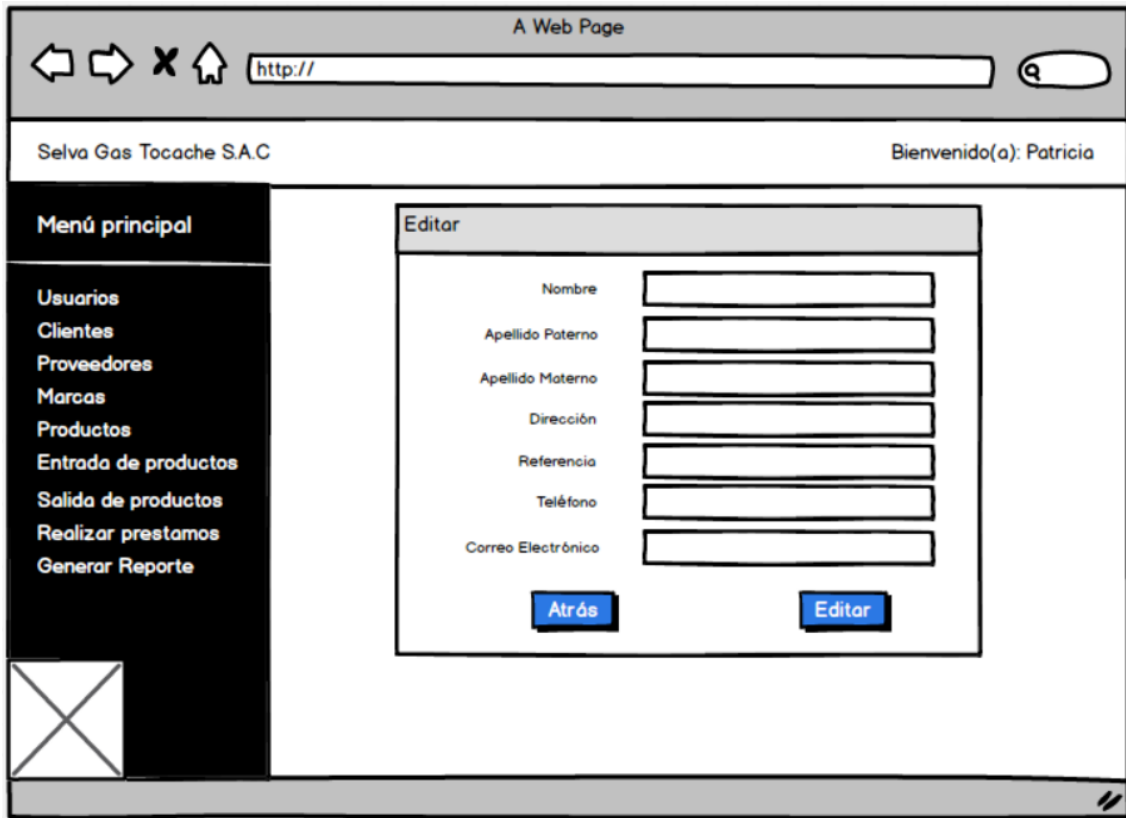


Figura N°65. Maquetación del módulo cliente - editar

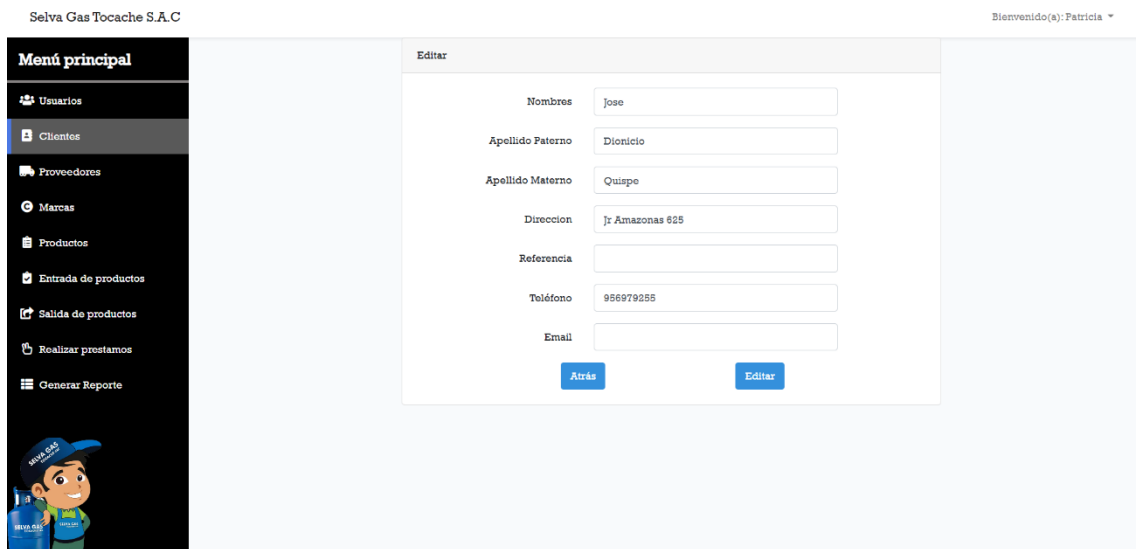


Figura N°66. Prototipo del módulo cliente - eliminar



Figura N°67. Maquetación del módulo cliente - eliminar

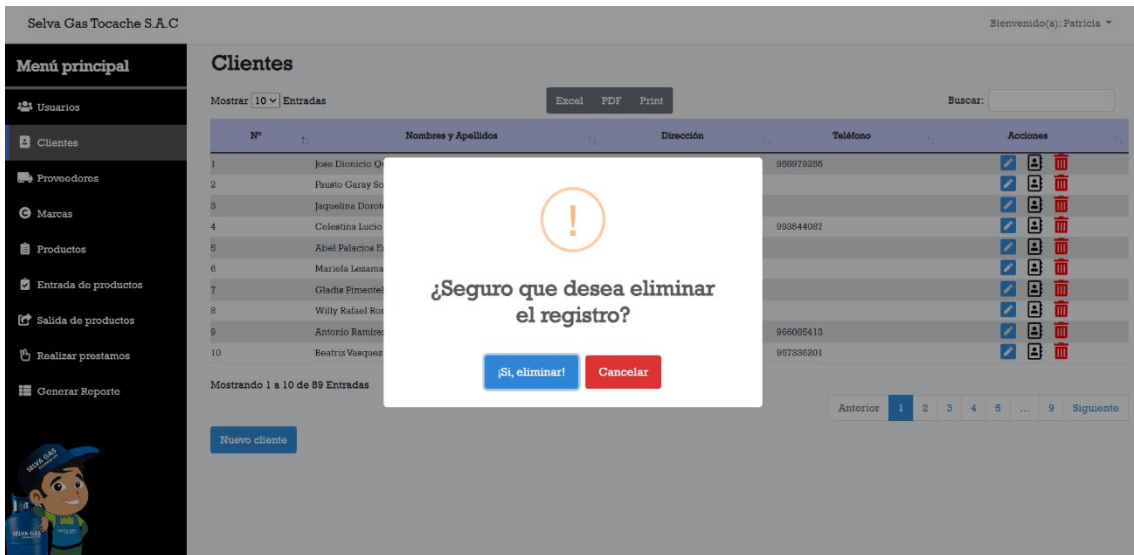


Figura N°68. Prototipo del módulo proveedores

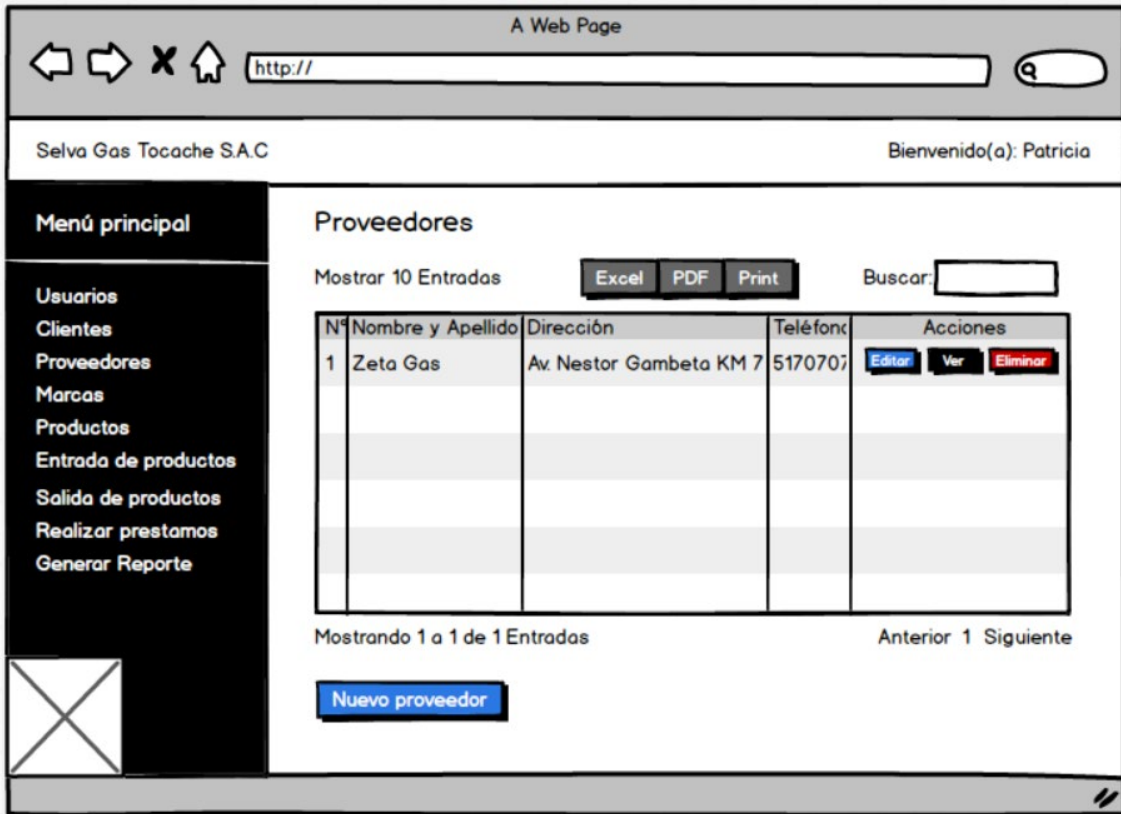


Figura N°69. Maquetación del módulo proveedores

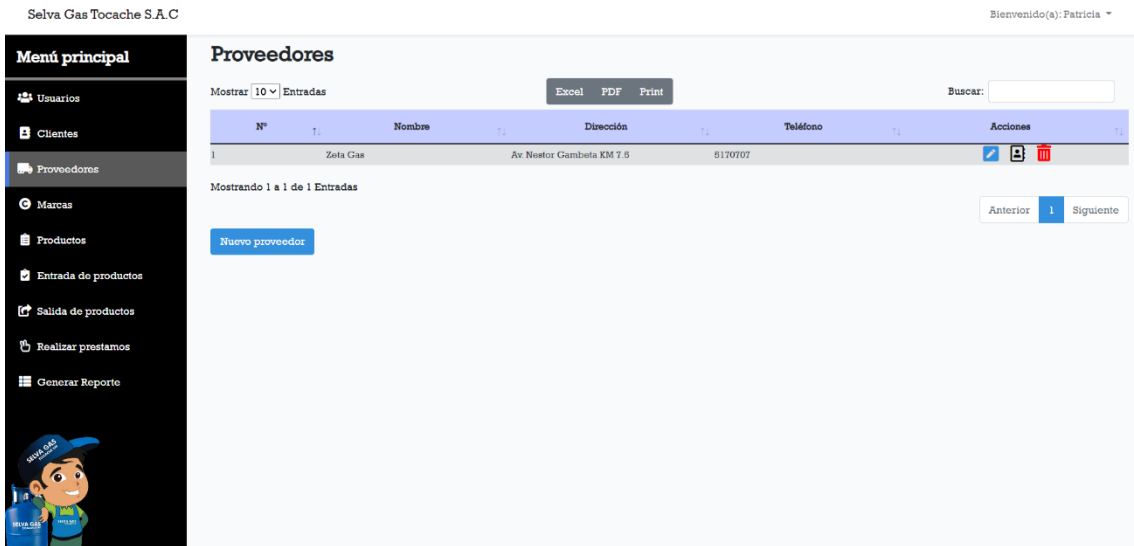


Figura N°70. Prototipo del módulo proveedores - registrar

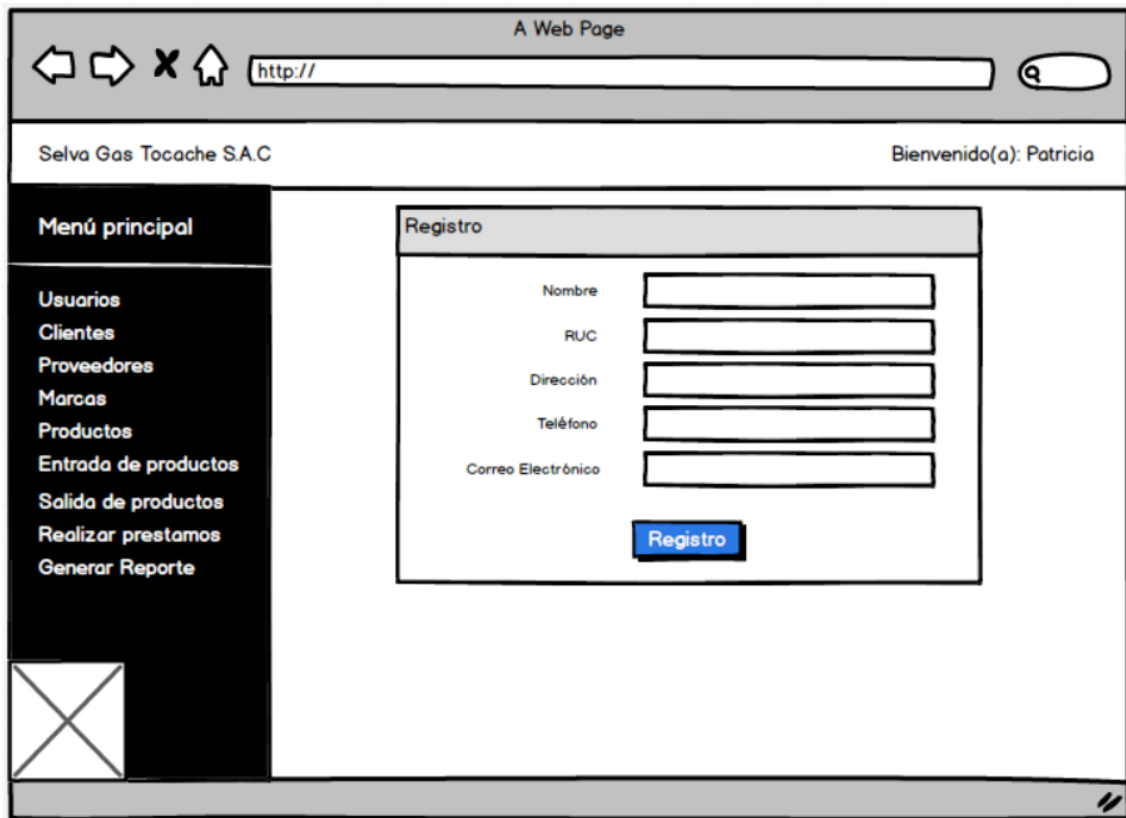


Figura N°71. Maquetación del módulo proveedores - registrar

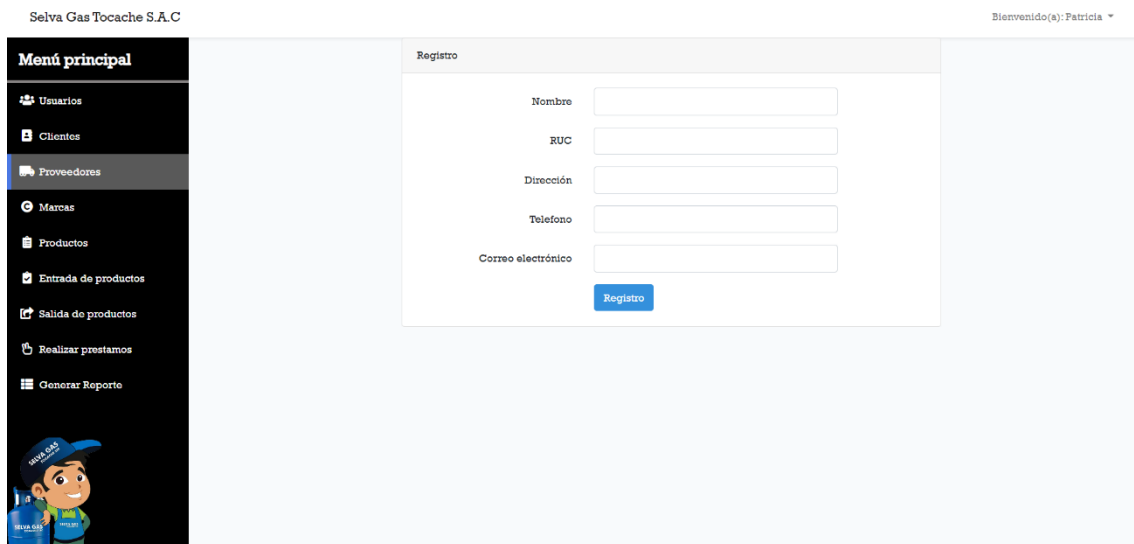


Figura N°72. Prototipo del módulo proveedores - editar

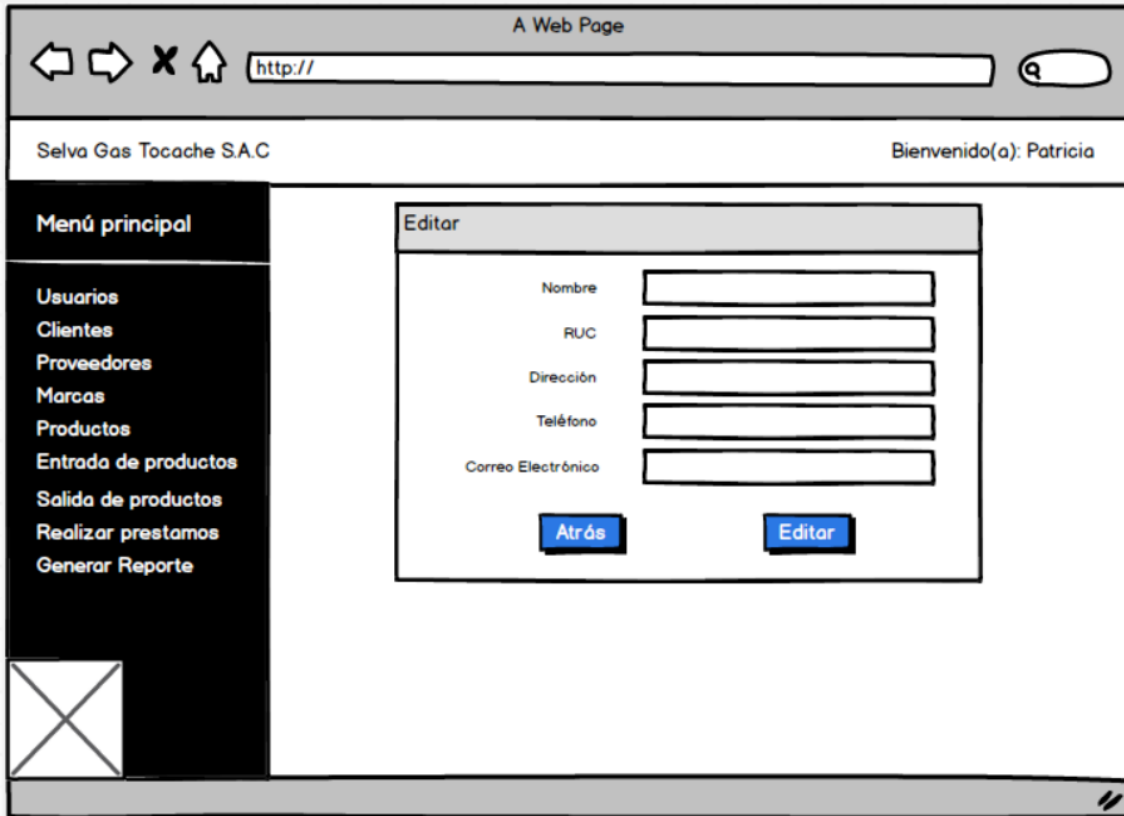


Figura N°73. Maquetación del módulo proveedores - editar

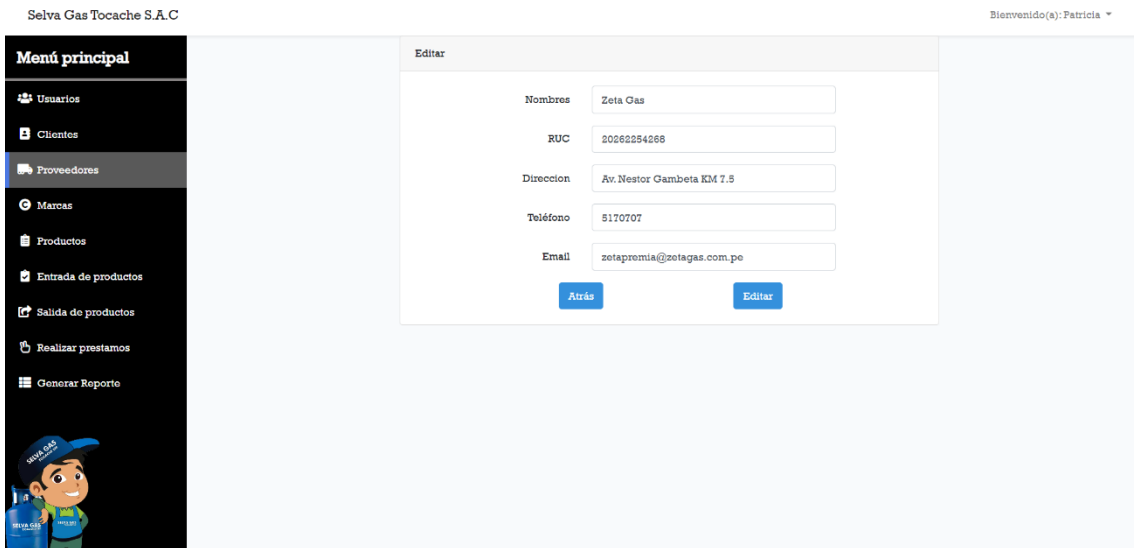


Figura N°74. Prototipo del módulo proveedores - eliminar



Figura N°75. Maquetación del módulo proveedores - eliminar

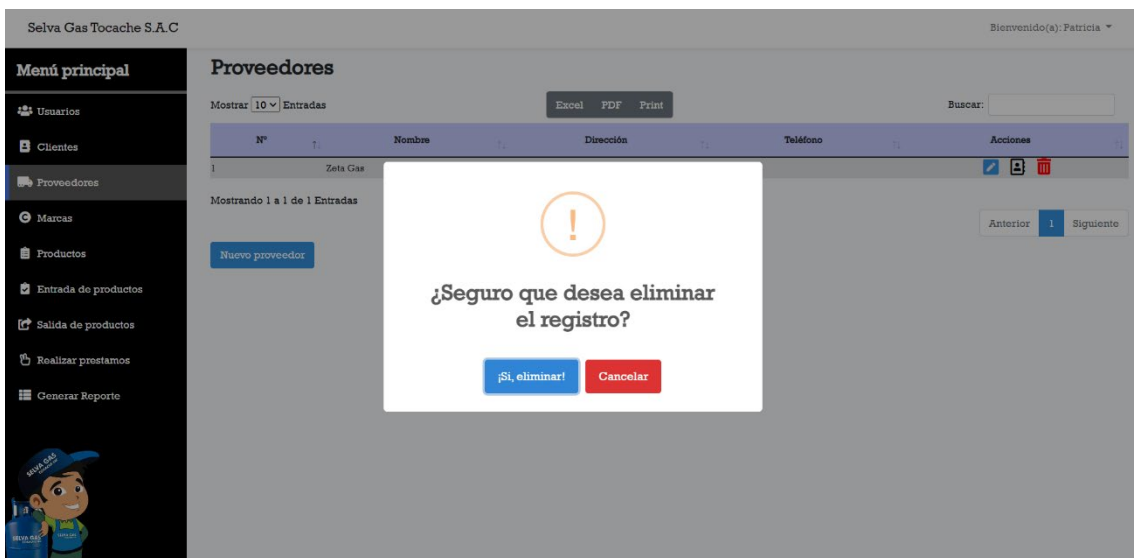




Figura N°76. Prototipo del módulo marcas

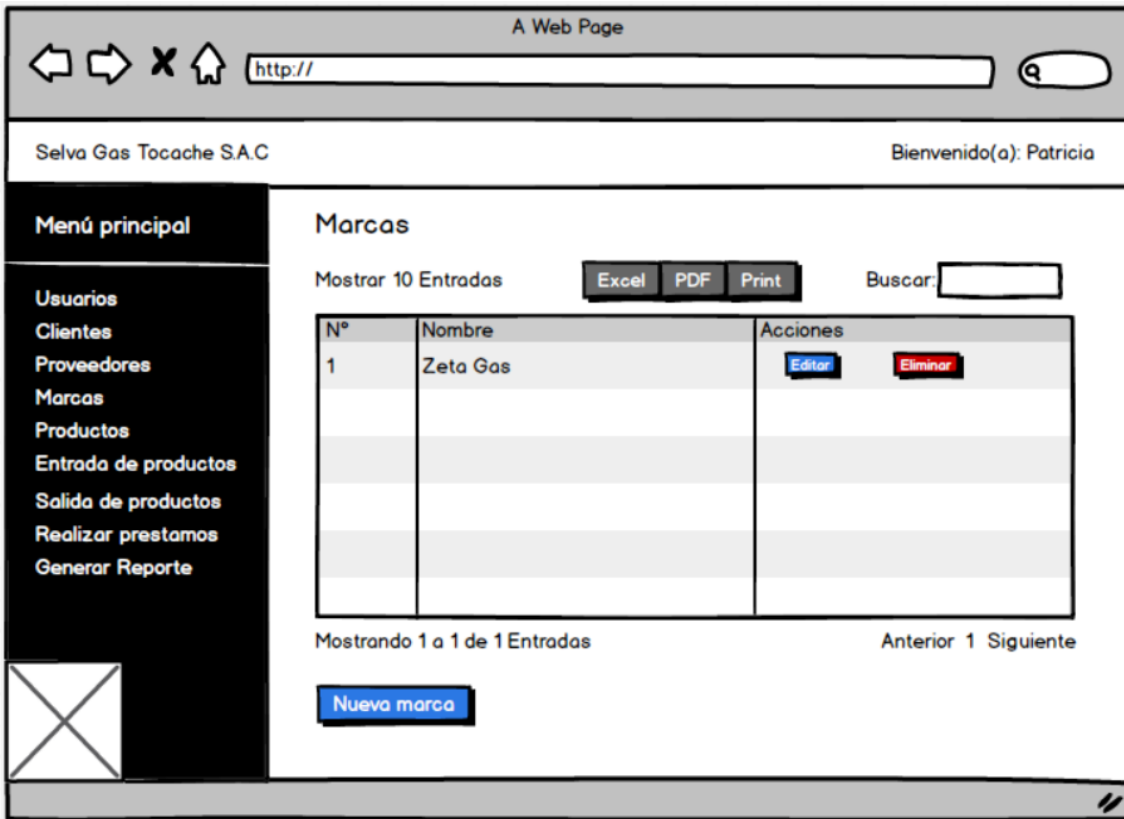


Figura N°77. Maquetación del módulo marcas

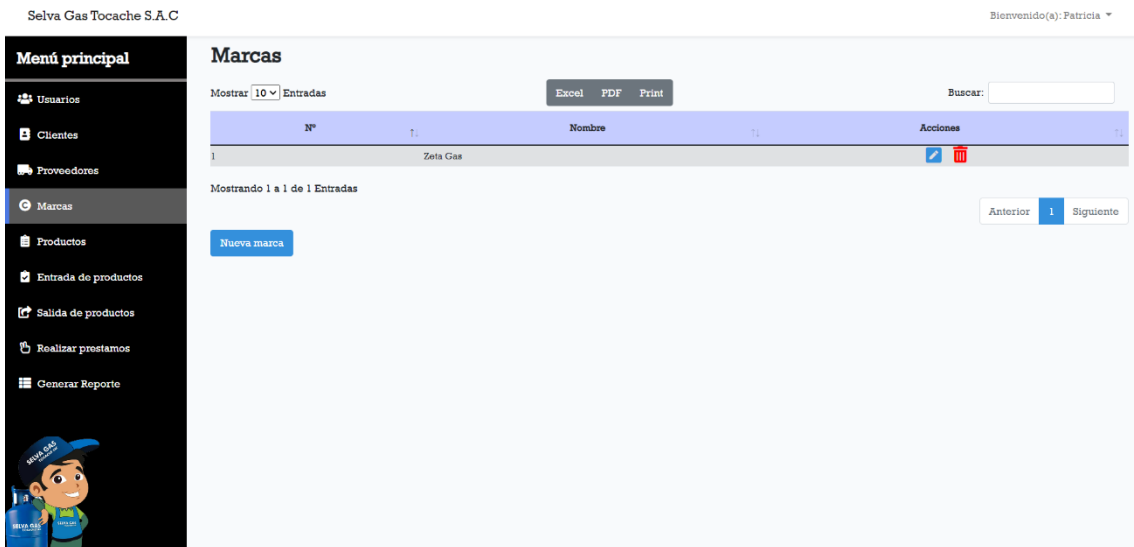


Figura N°78. Prototipo del módulo marcas - registrar

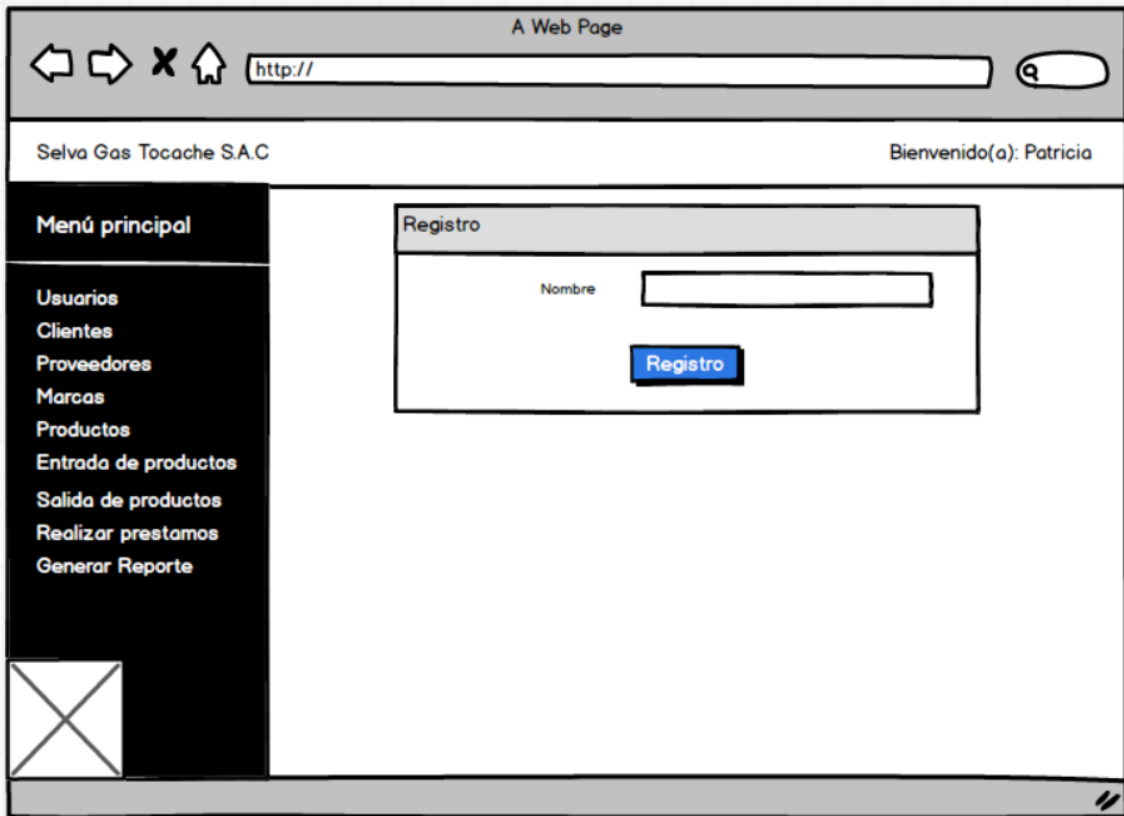


Figura N°79. Maquetación del módulo marcas - registrar

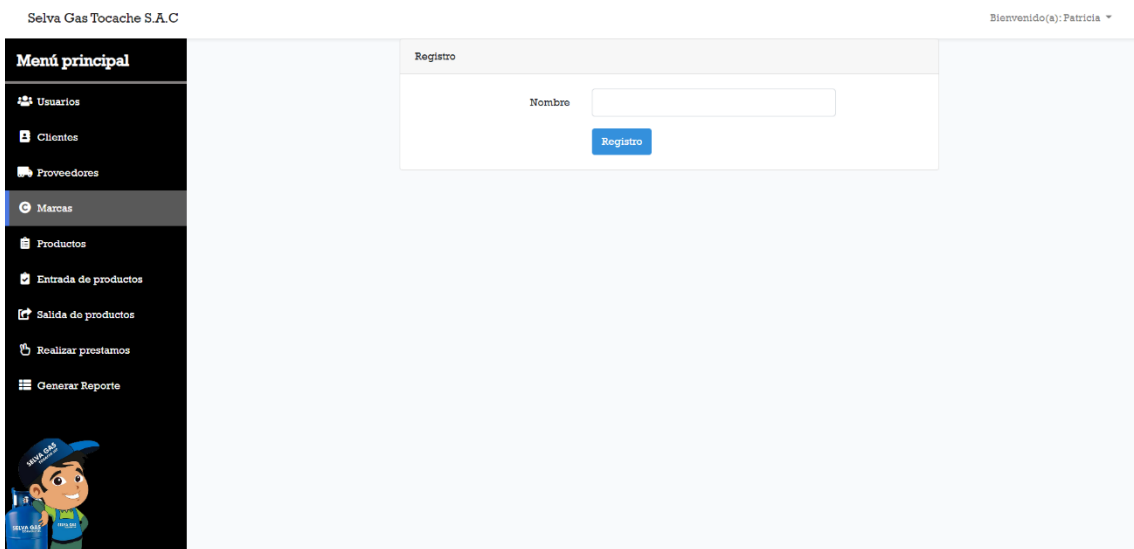


Figura N°80. Prototipo del módulo marcas - editar

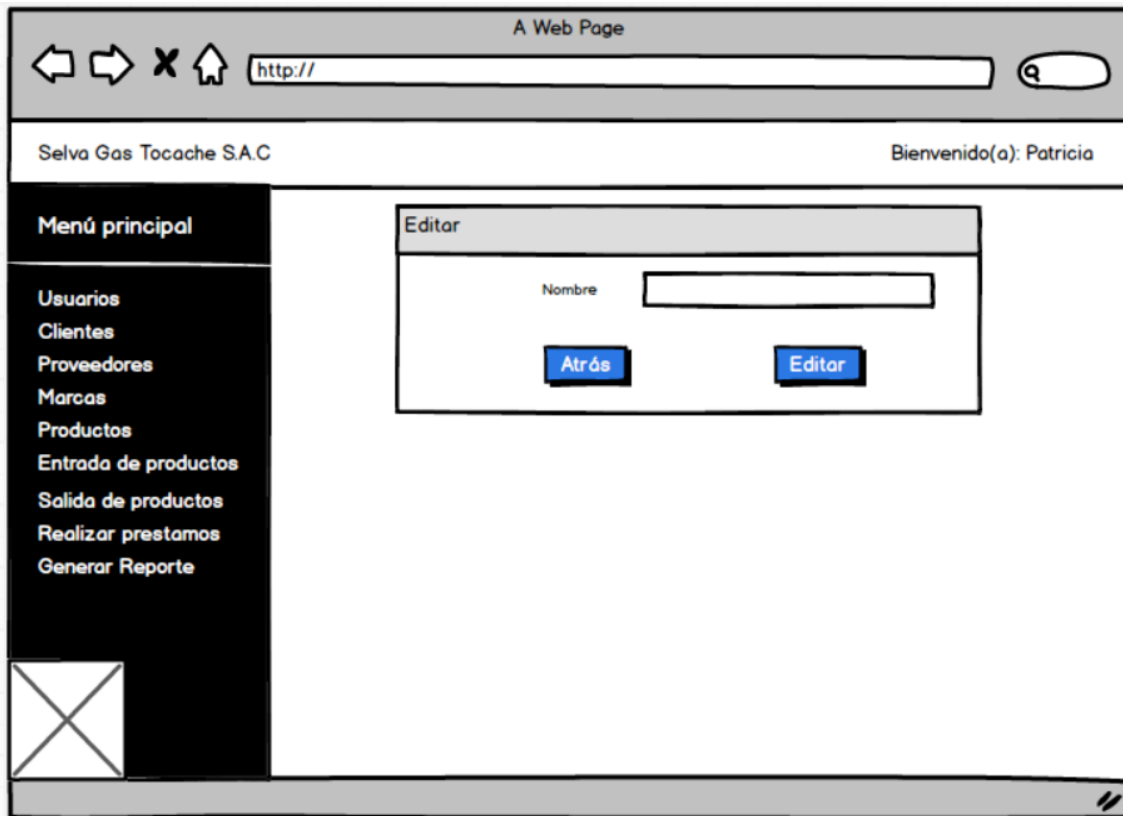


Figura N°81. Maquetación del módulo marcas - editar

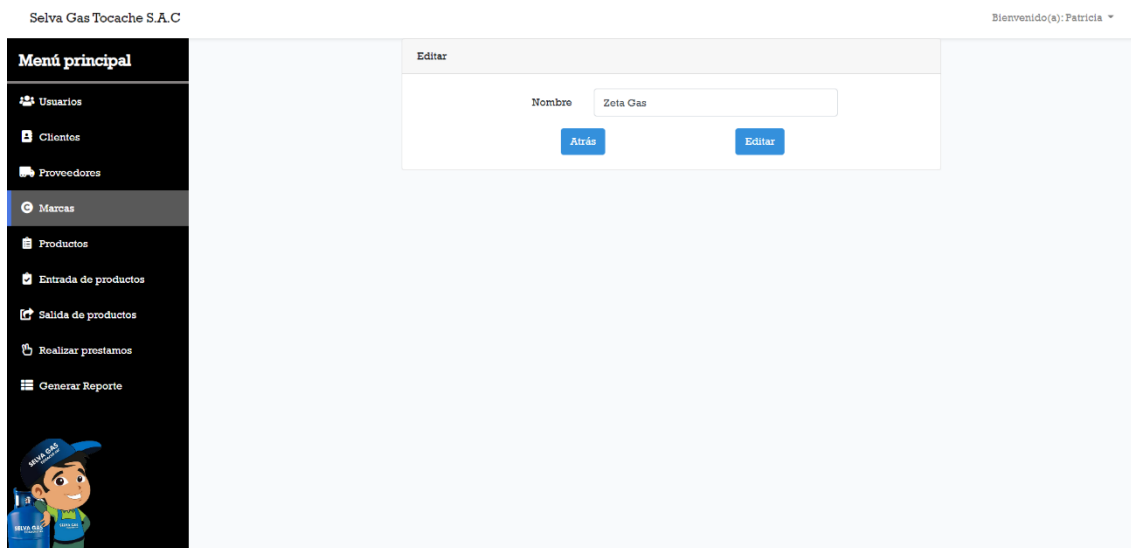


Figura N°82. Prototipo del módulo marcas - eliminar



Figura N°83. Maquetación del módulo marcas - eliminar

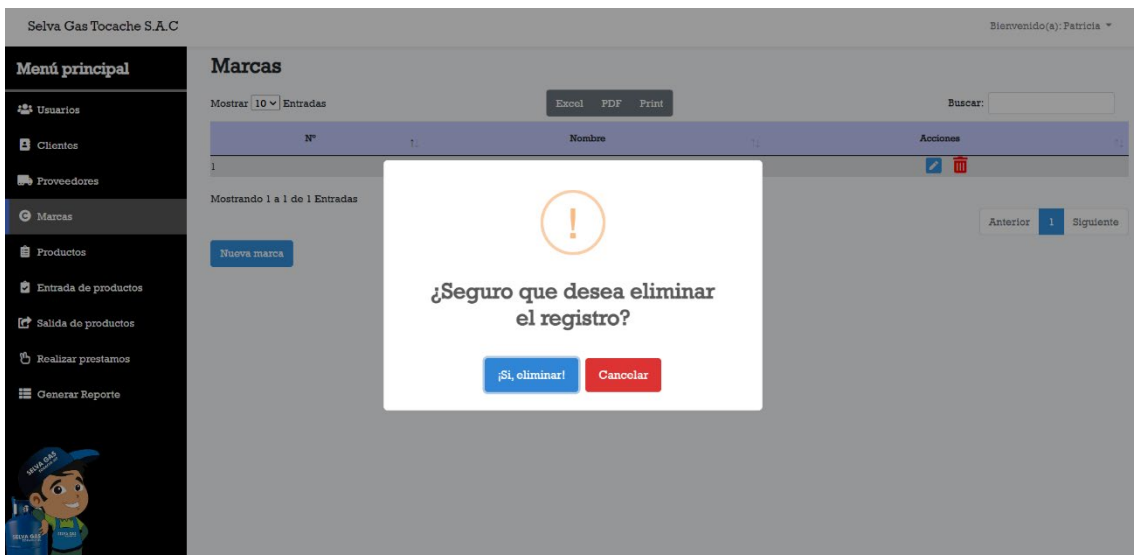


Figura N°84. Prototipo del módulo productos

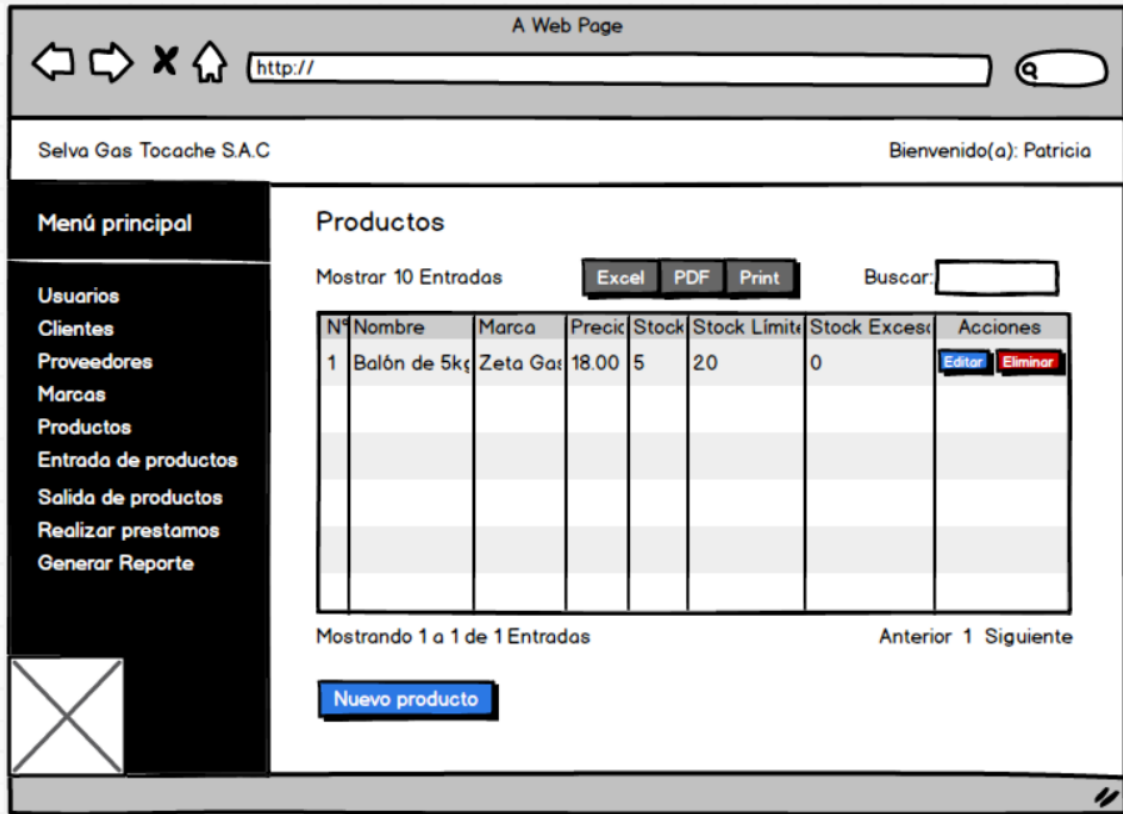


Figura N°85. Maquetación del módulo productos

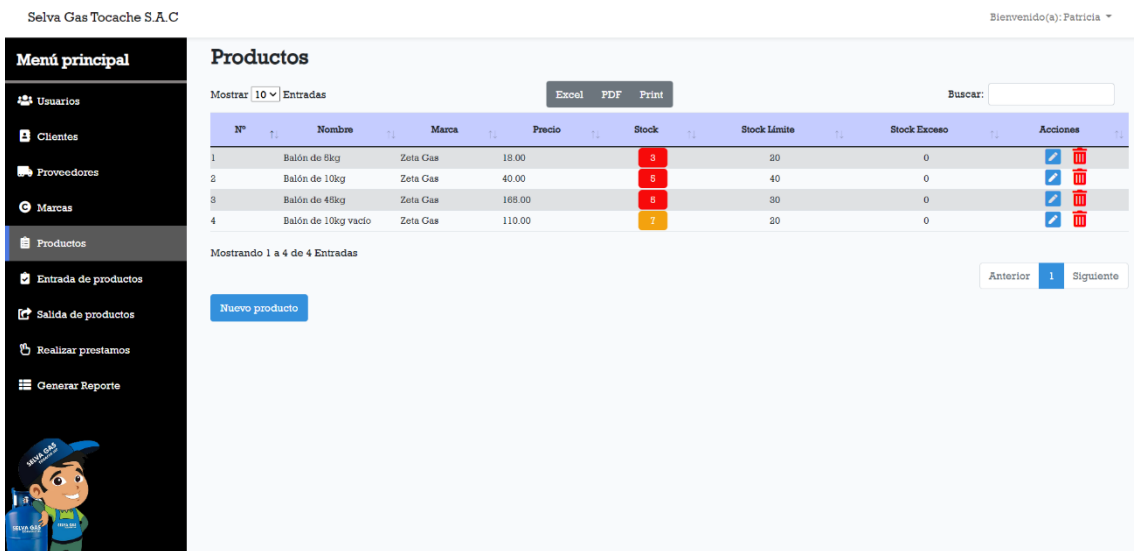


Figura N°86. Prototipo del módulo productos - registrar

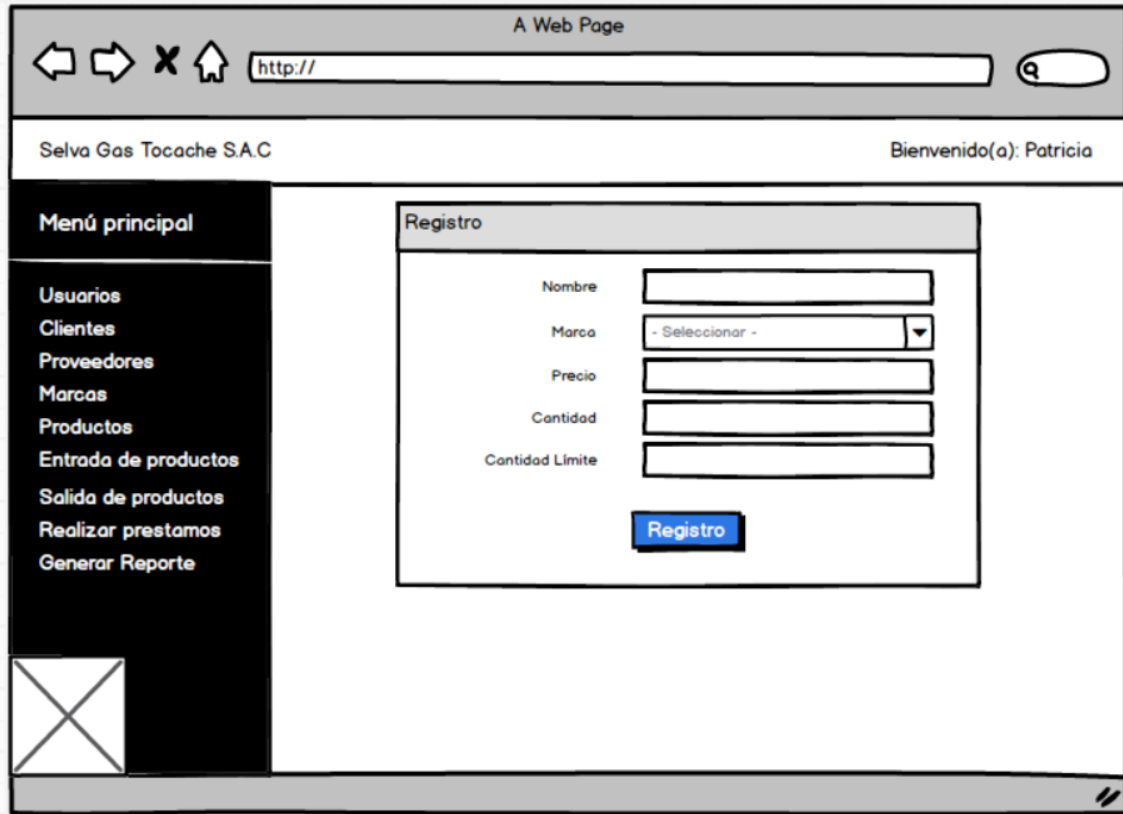


Figura N°87. Maquetación del módulo productos - registrar

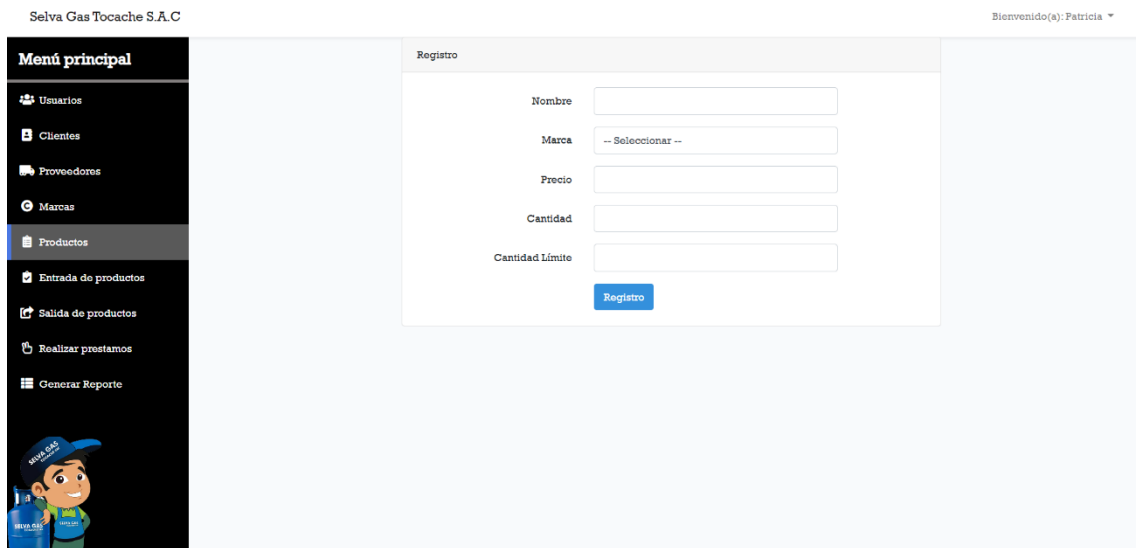


Figura N°88. Prototipo del módulo productos - editar

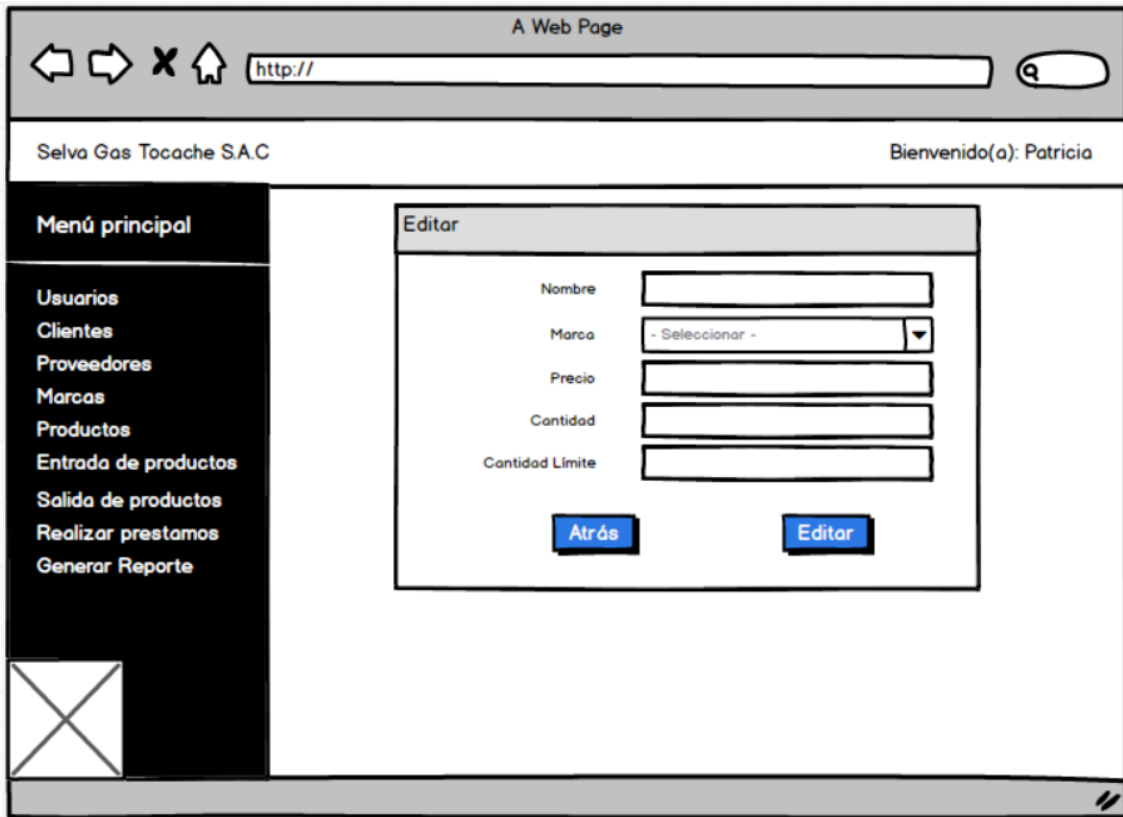


Figura N°89. Maquetación del módulo productos - editar

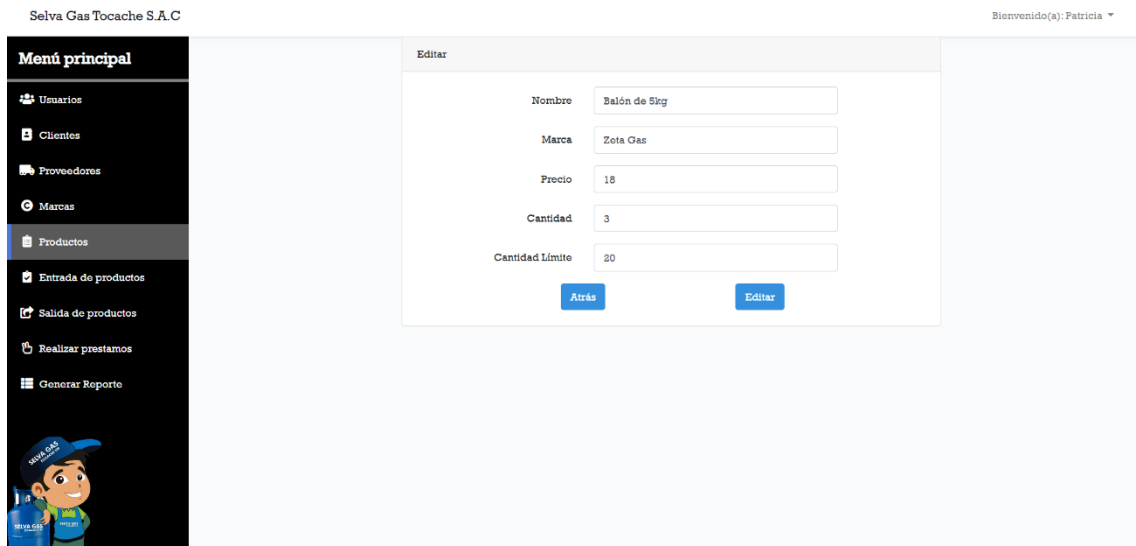


Figura N°90. Prototipo del módulo productos - eliminar



Figura N°91. Maquetación del módulo productos - eliminar

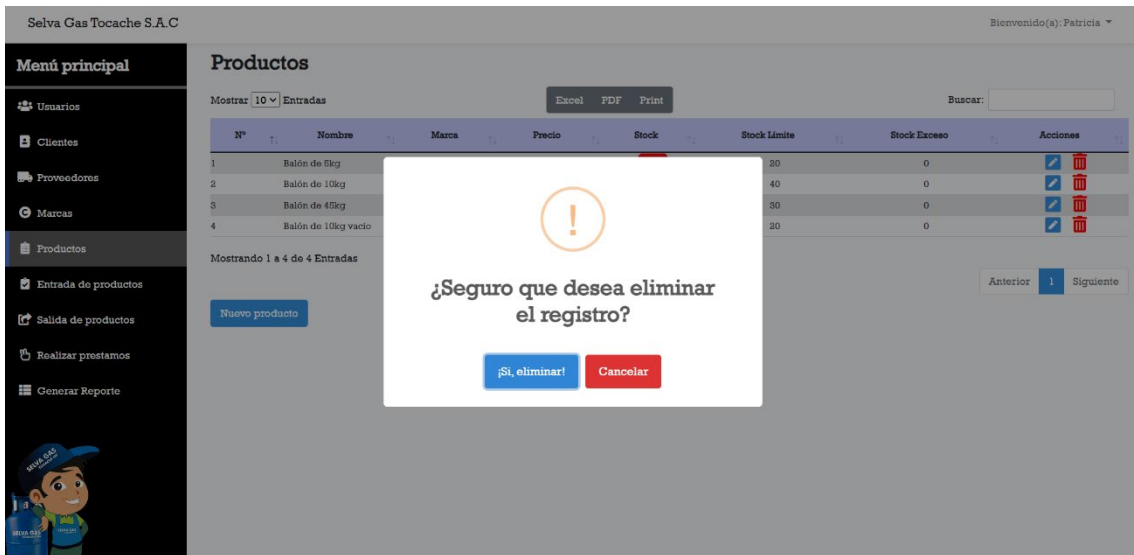




Figura N°92. Prototipo del módulo entrada de productos



Figura N°93. Maquetación del módulo entrada de productos

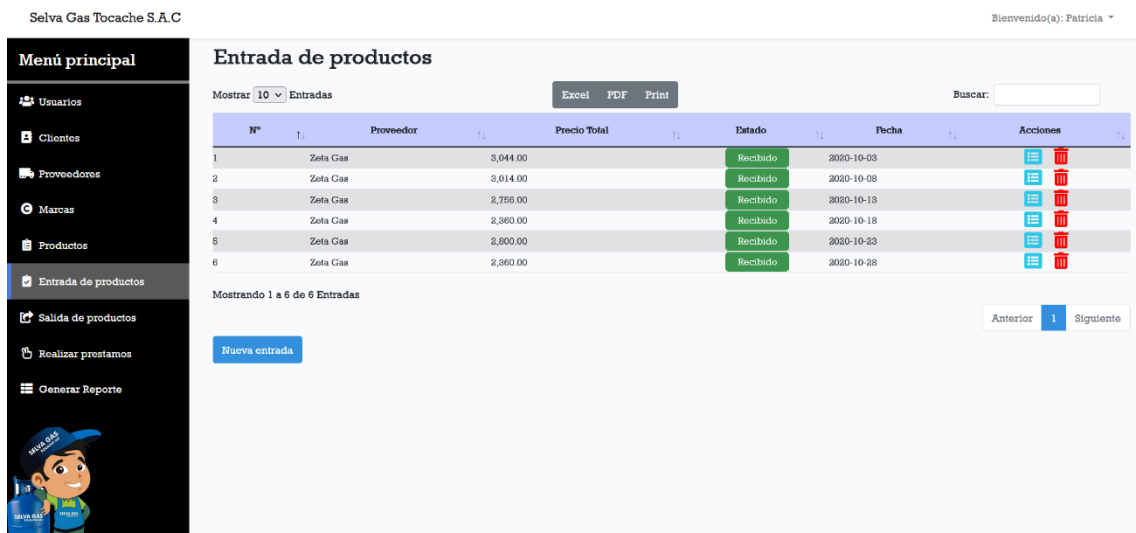


Figura N°94. Prototipo del módulo entrada de productos - registrar

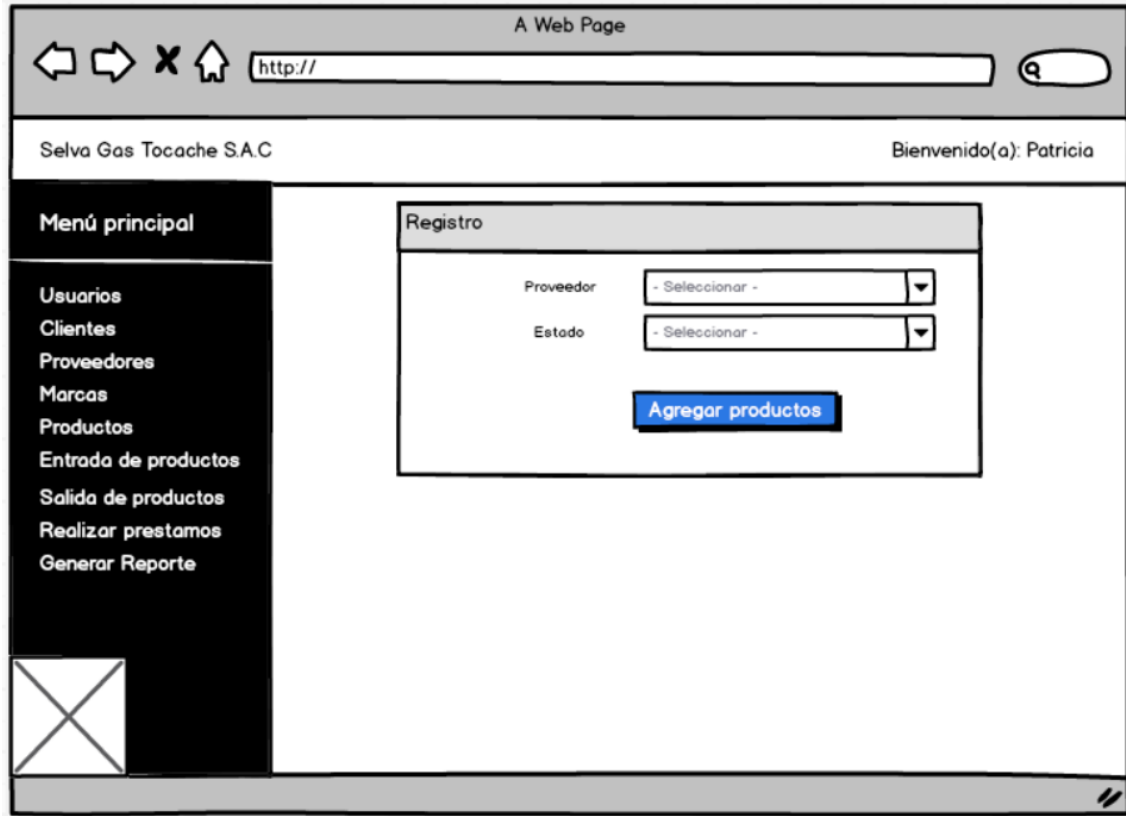


Figura N°95. Maquetación del módulo entrada de productos - registrar

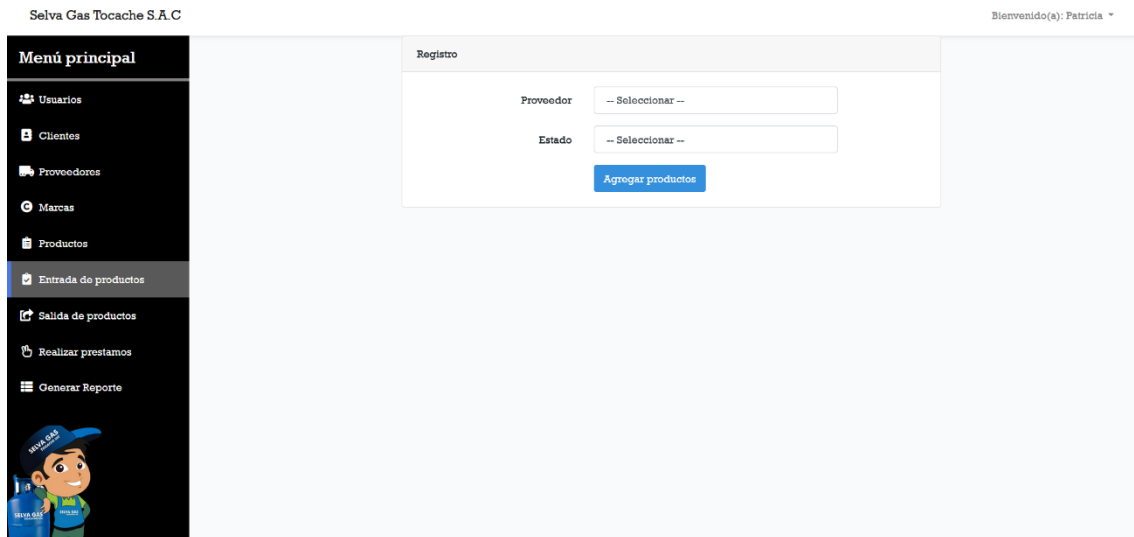


Figura N°96. Prototipo del módulo entrada de productos – lista de productos

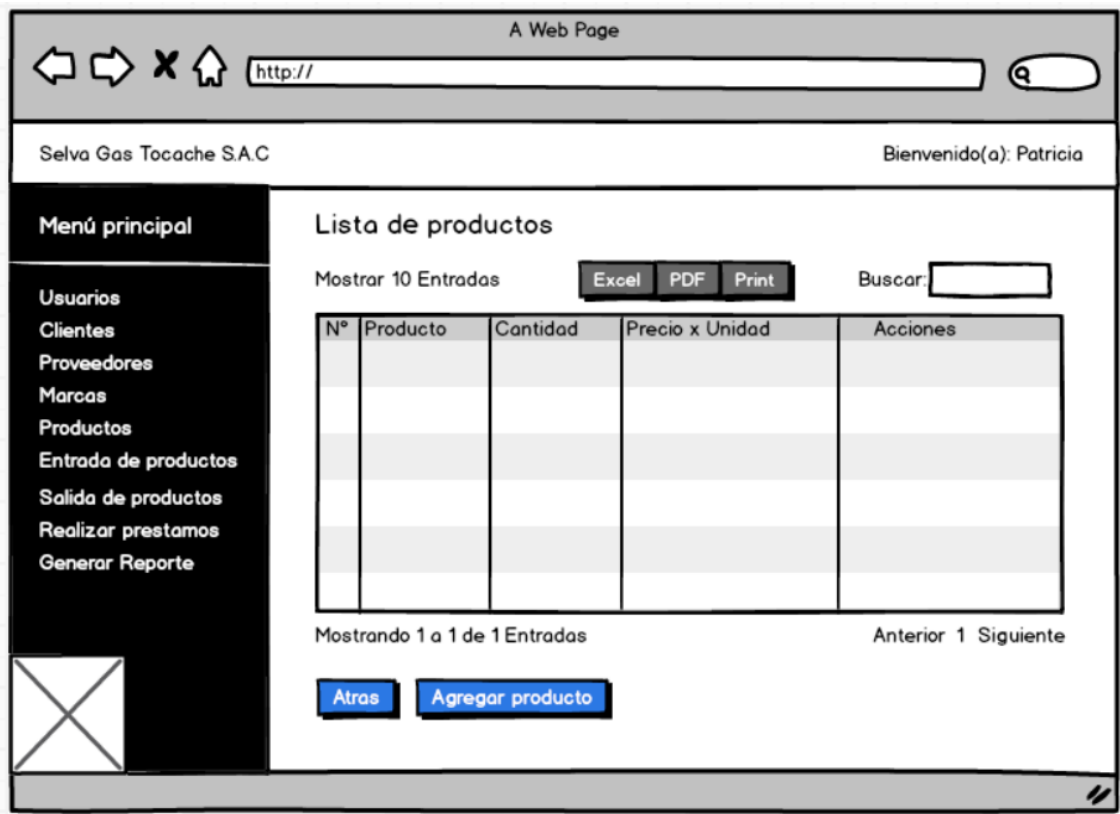


Figura N°97. Maquetación del módulo entrada de productos – lista de productos

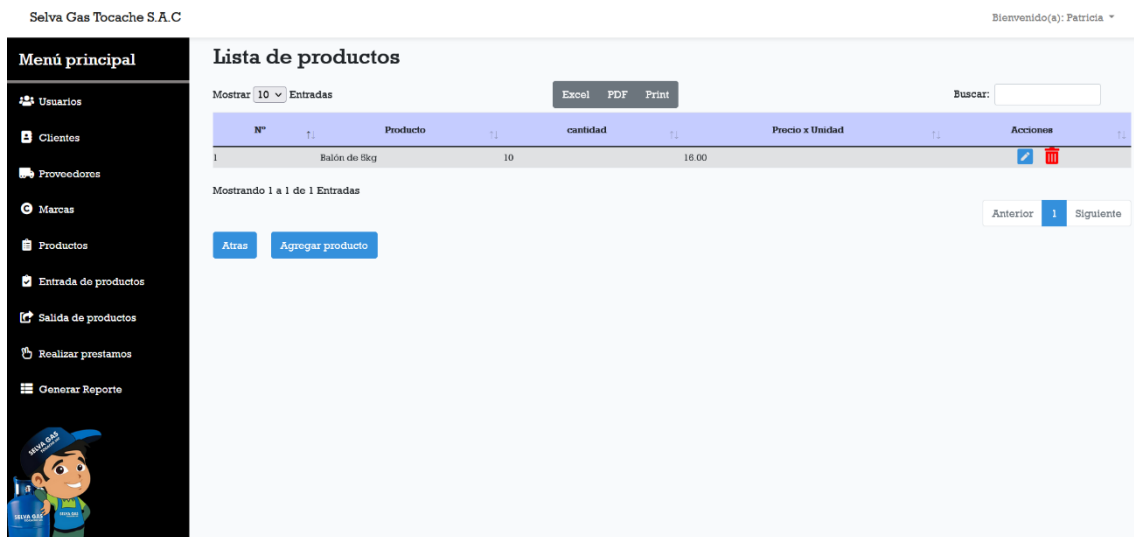


Figura N°98. Prototipo del módulo entrada de productos – agregar producto

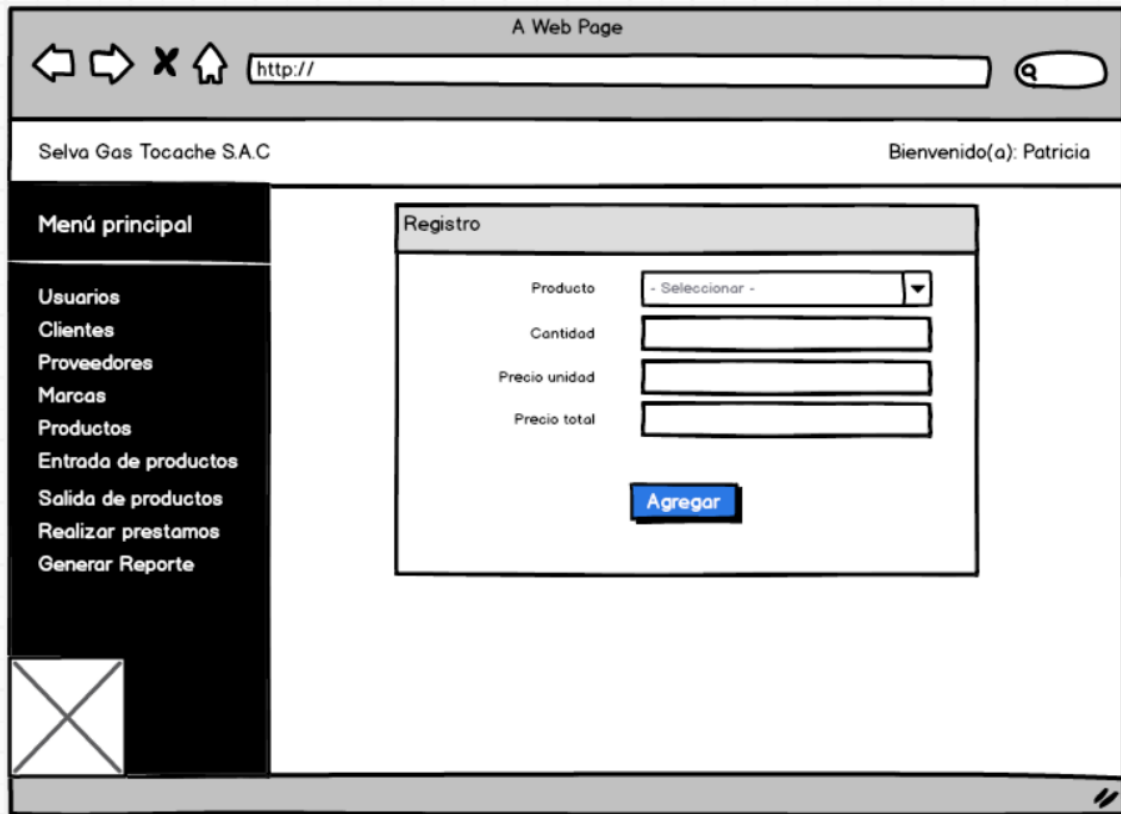


Figura N°99. Maquetación del módulo entrada de productos – agregar producto

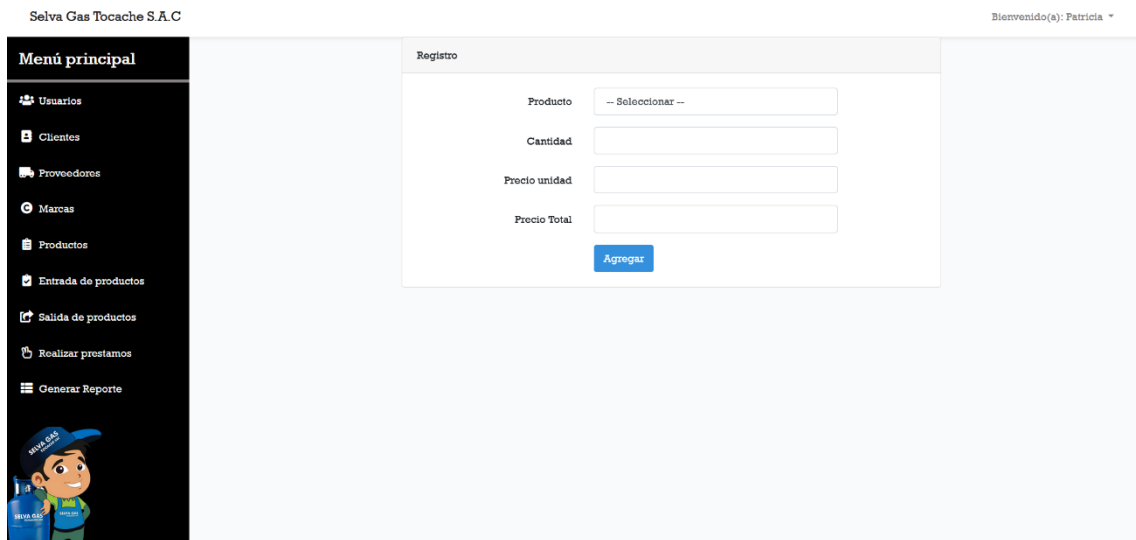


Figura N°100. Prototipo del módulo entrada de productos – editar

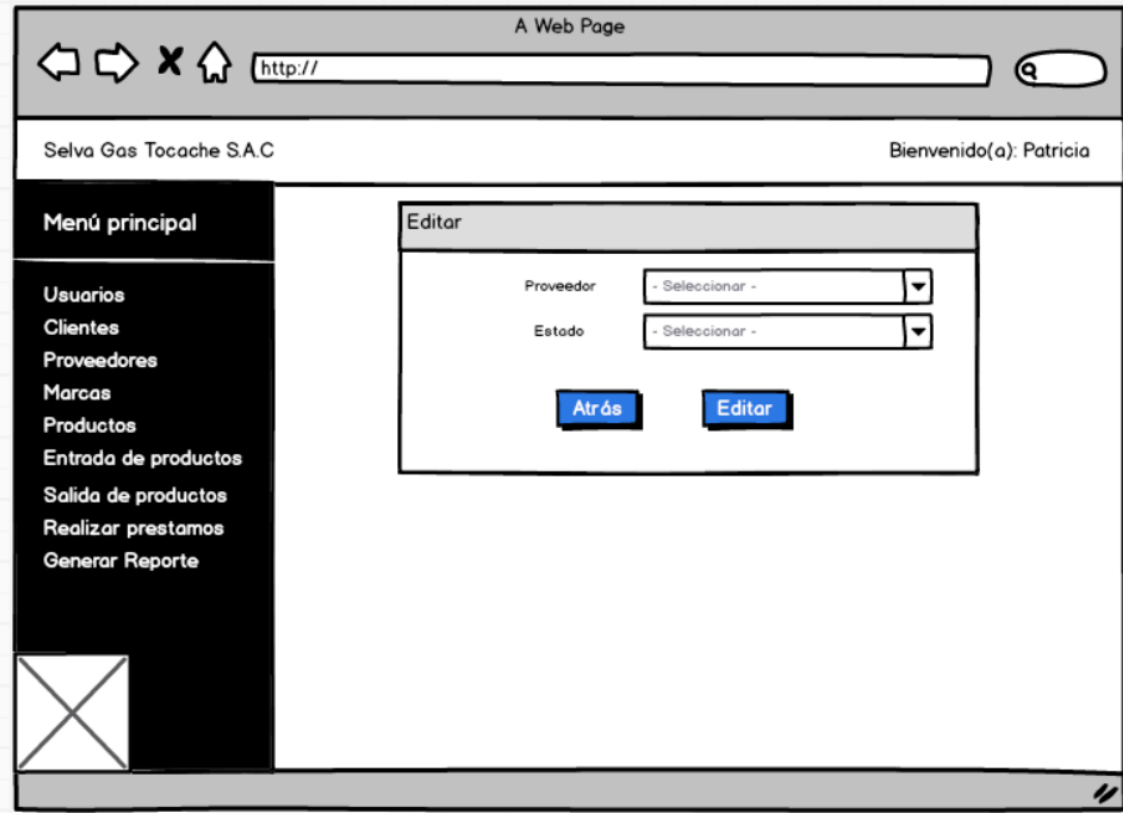


Figura N°101. Maquetación del módulo entrada de productos – editar

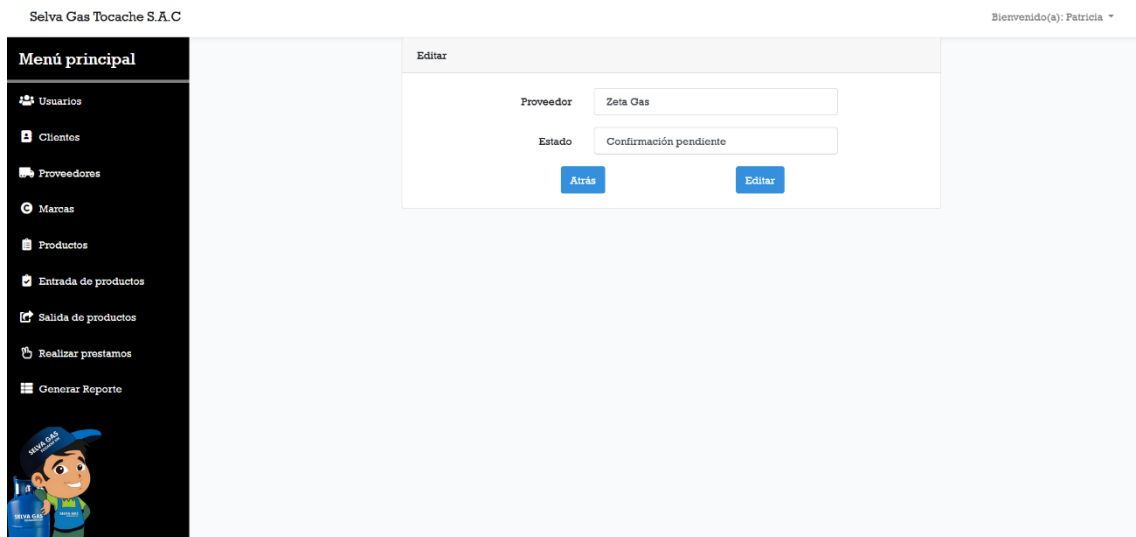


Figura N°102. Prototipo del módulo entrada de productos – eliminar



Figura N°103. Maquetación del módulo entrada de productos – eliminar

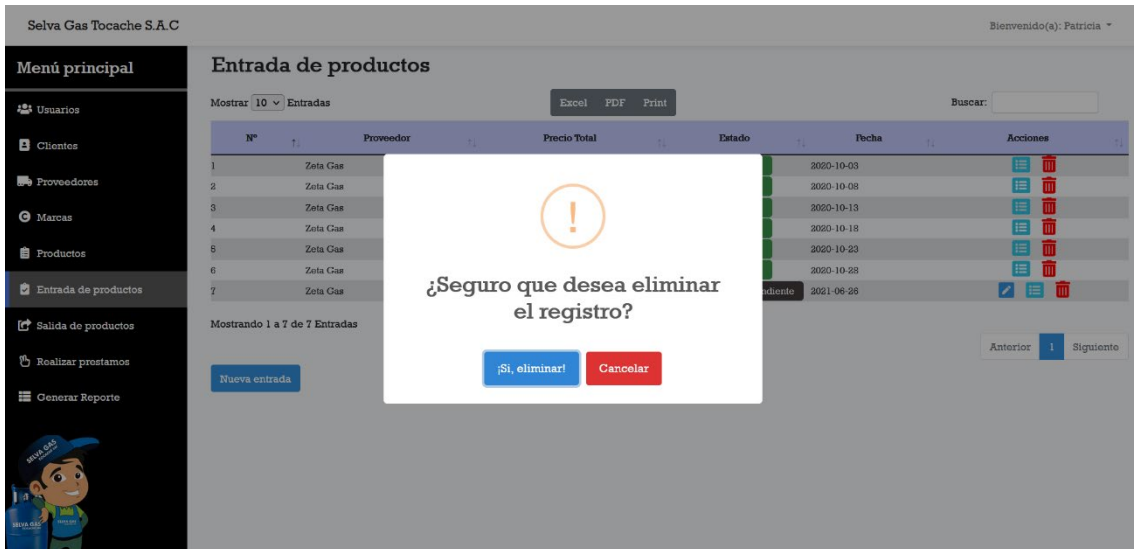


Figura N°104. Prototipo del módulo salida de productos



Figura N°105. Maquetación del módulo salida de productos

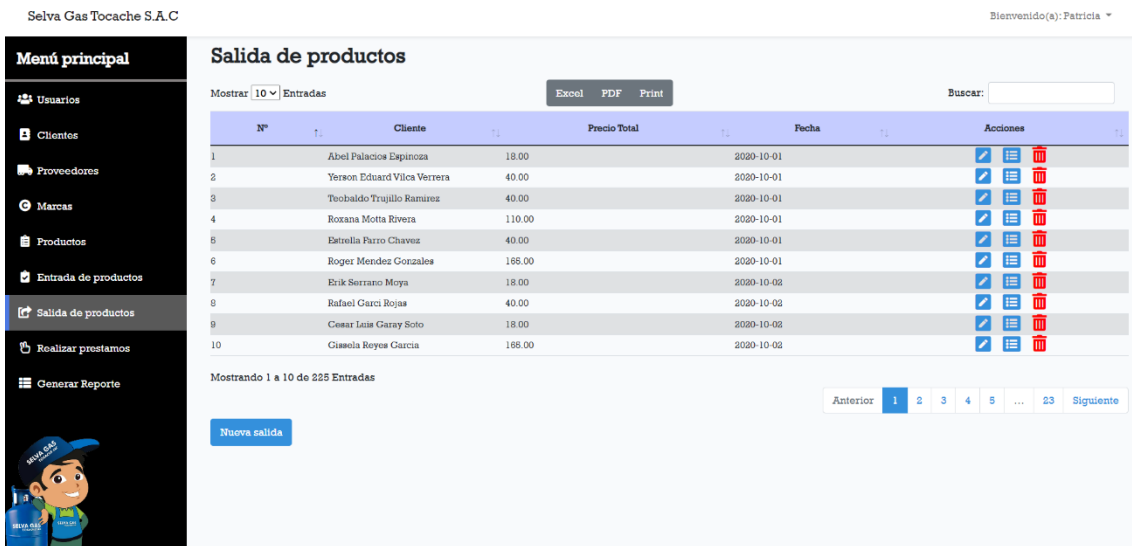


Figura N°106. Prototipo del módulo salida de productos - registrar

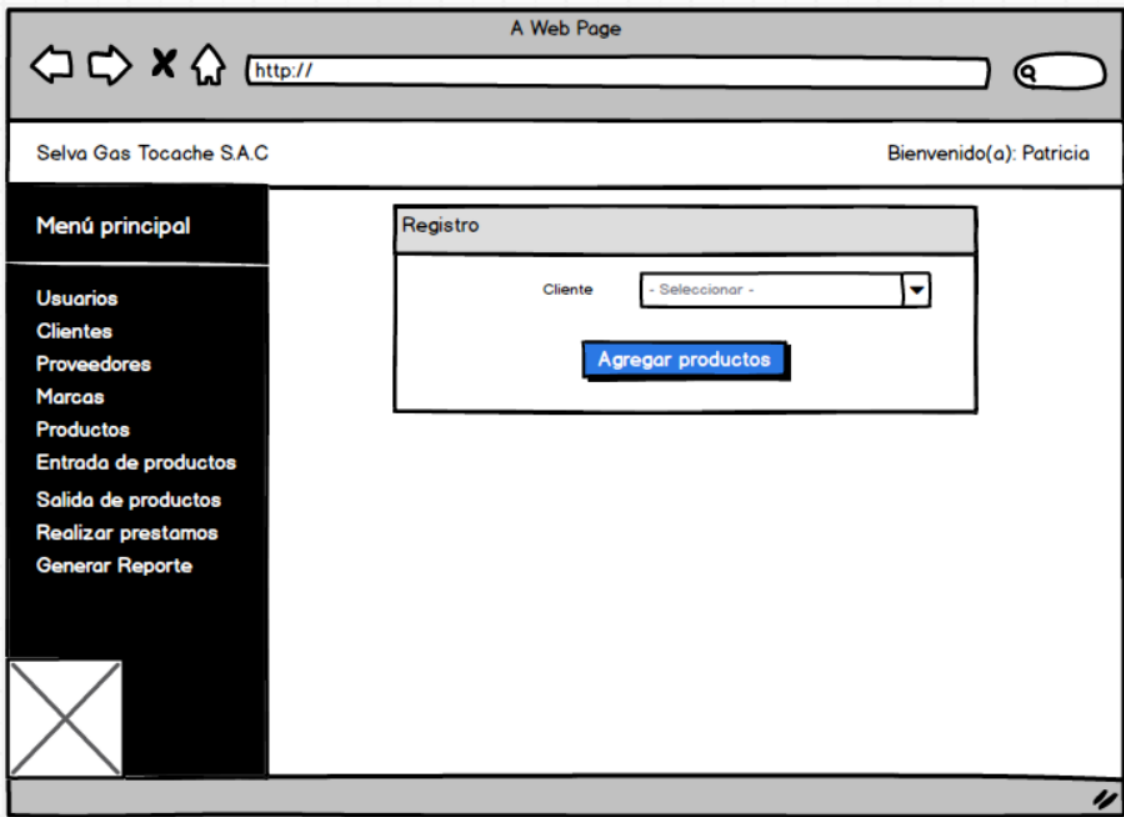


Figura N°107. Maquetación del módulo salida de productos - registrar

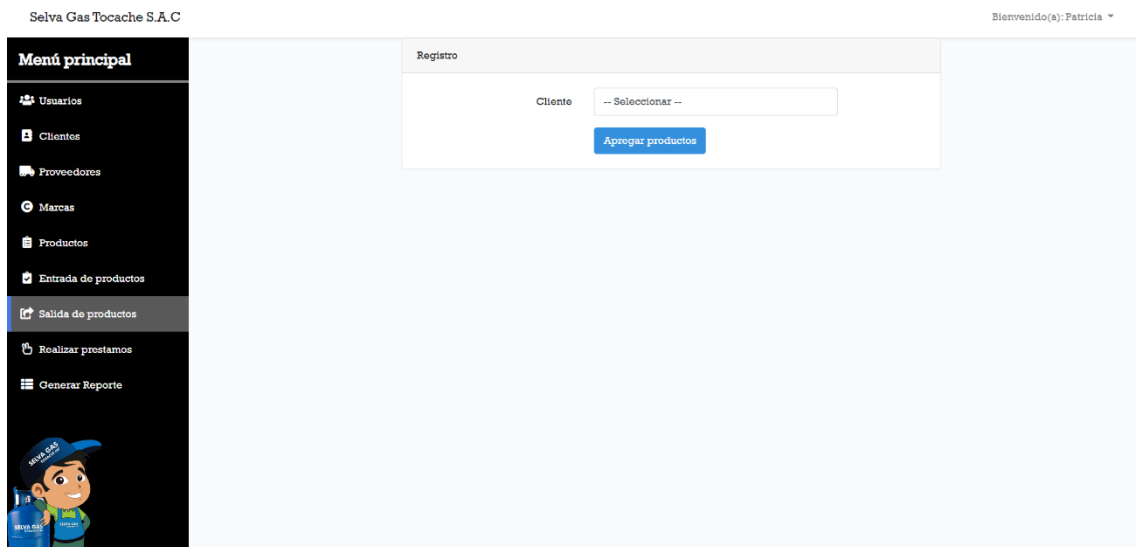




Figura N°108. Prototipo del módulo salida de productos – lista de productos



Figura N°109. Maquetación del módulo salida de productos – lista de productos

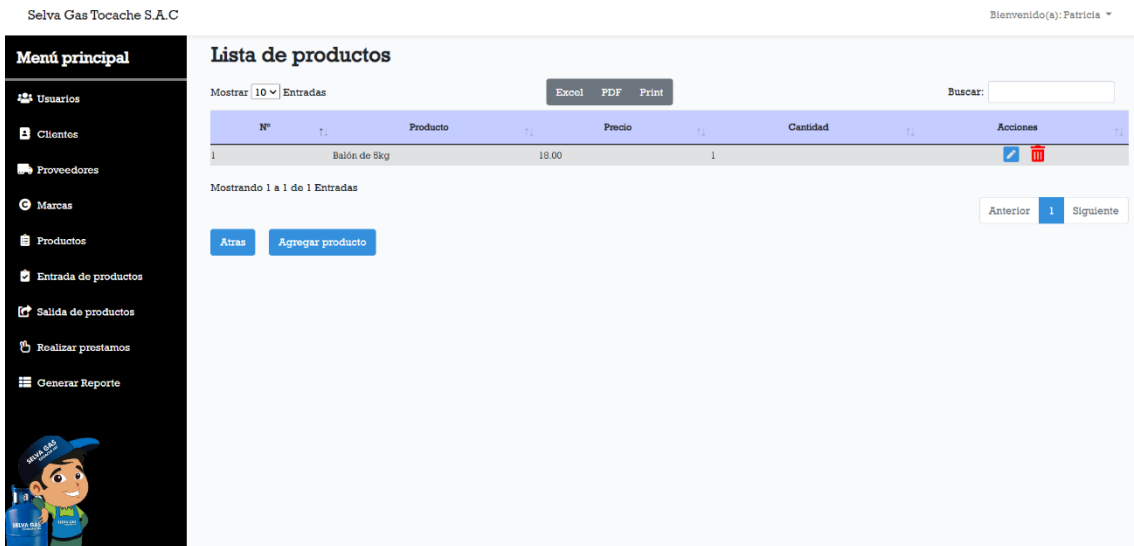


Figura N°110. Prototipo del módulo salida de productos – agregar producto

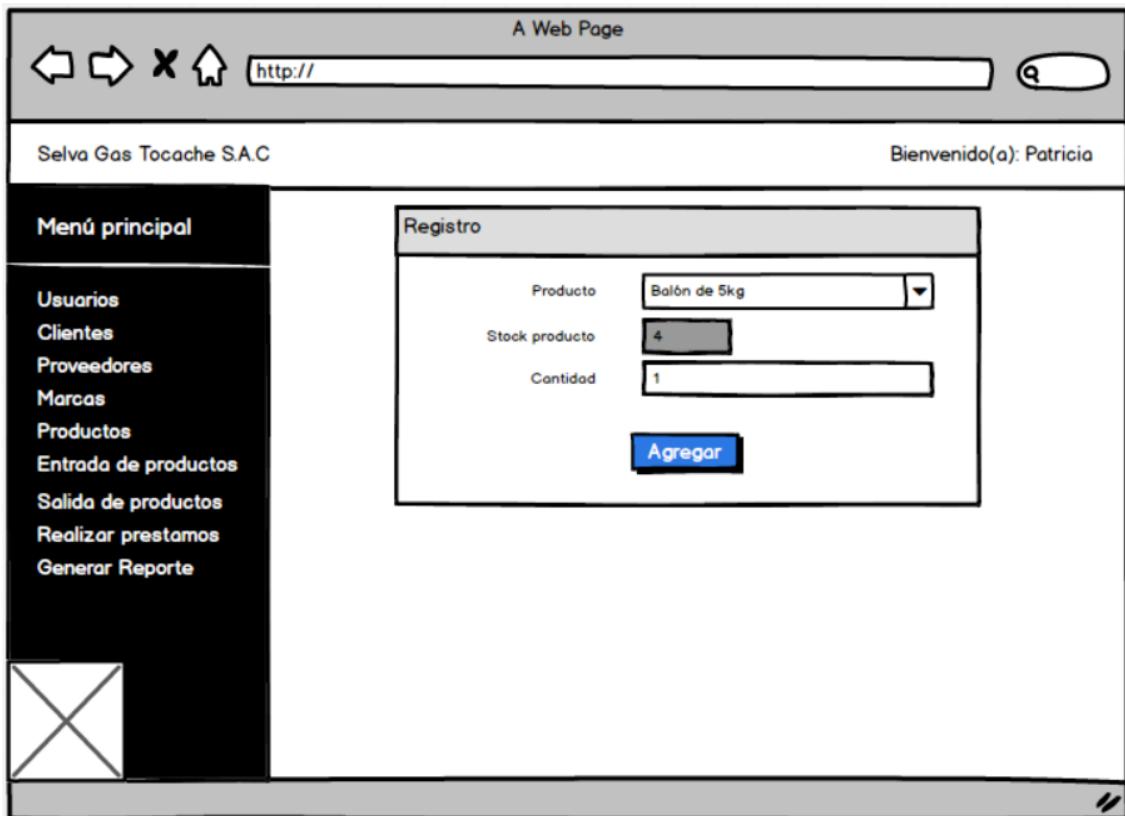


Figura N°111. Maquetación del módulo salida de productos – agregar producto

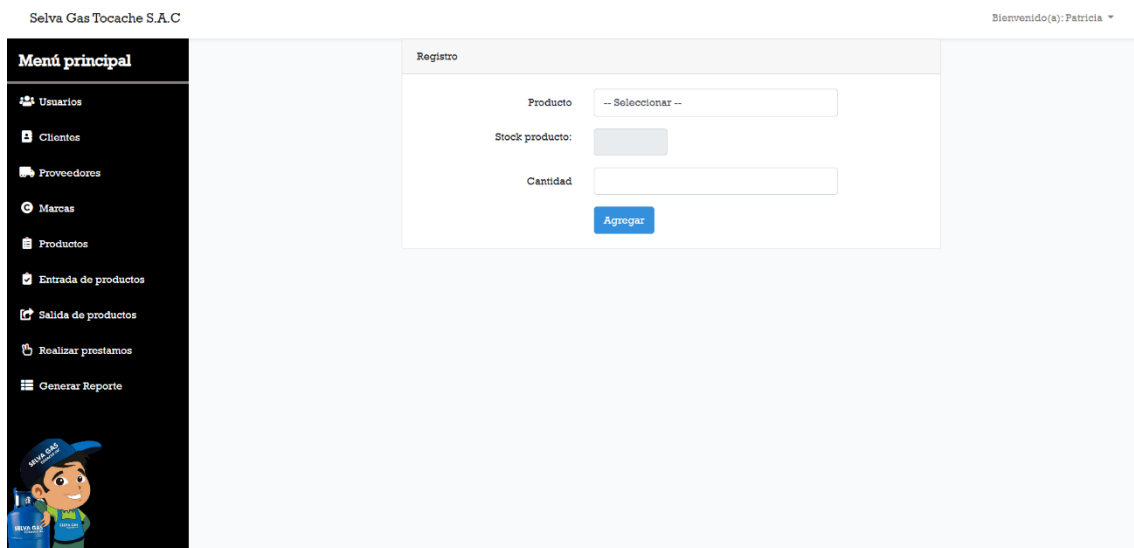


Figura N°112. Prototipo del módulo salida de productos – editar

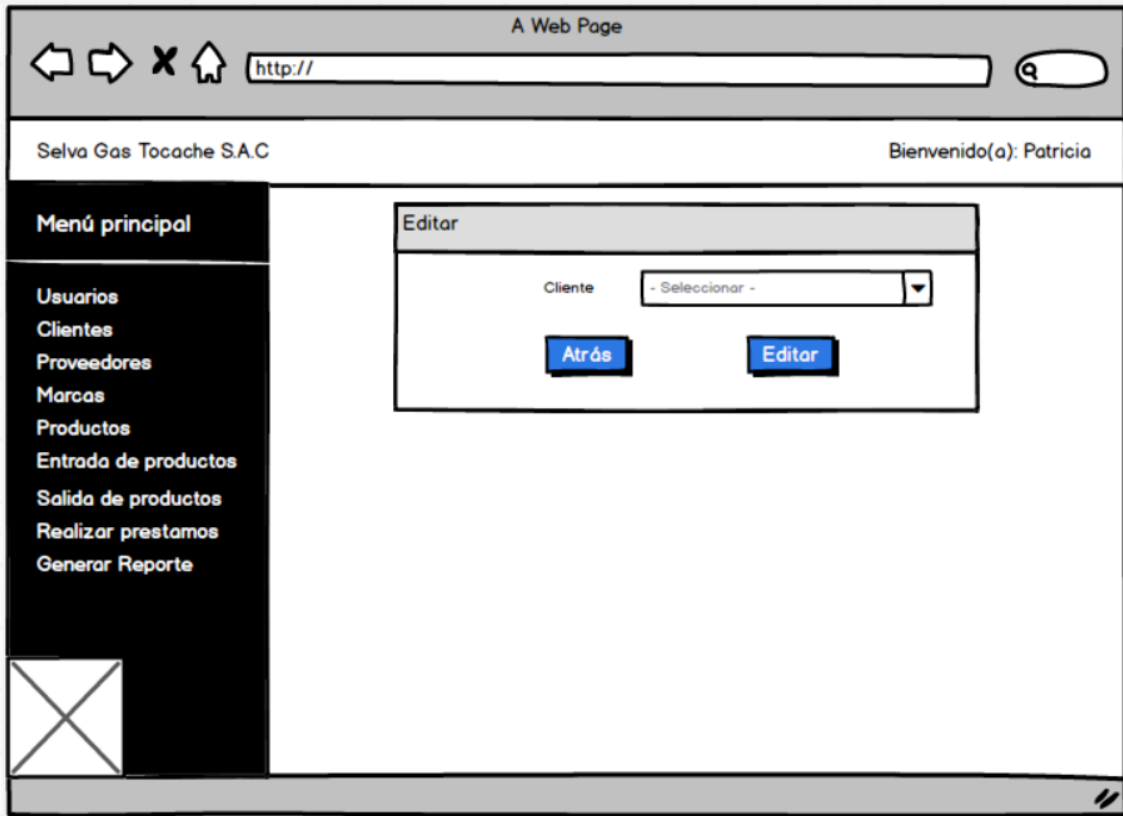


Figura N°113. Maquetación del módulo salida de productos – editar

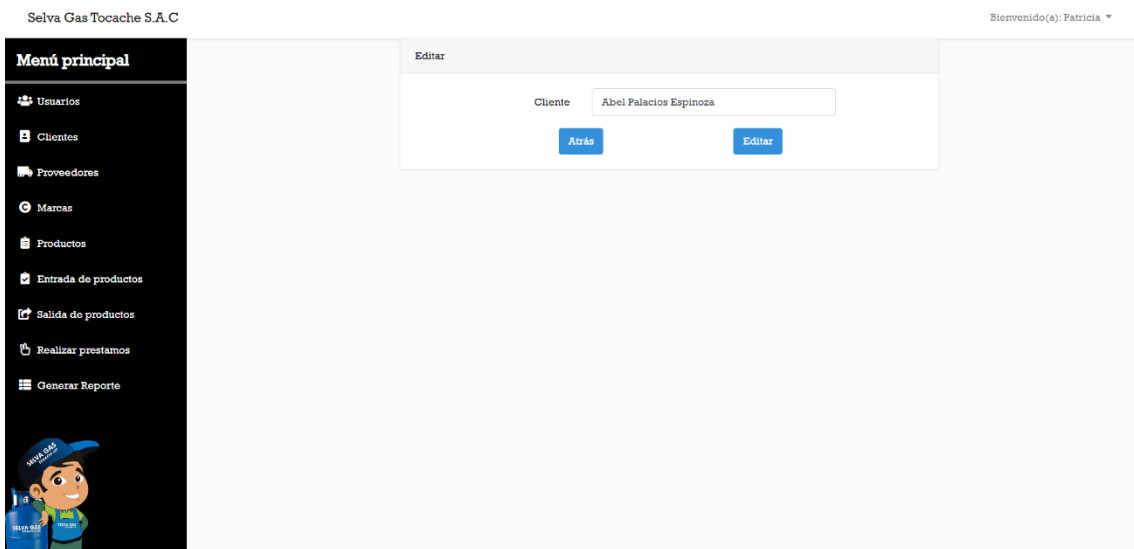


Figura N°114. Prototipo del módulo salida de productos – eliminar



Figura N°115. Maquetación del módulo salida de productos – eliminar

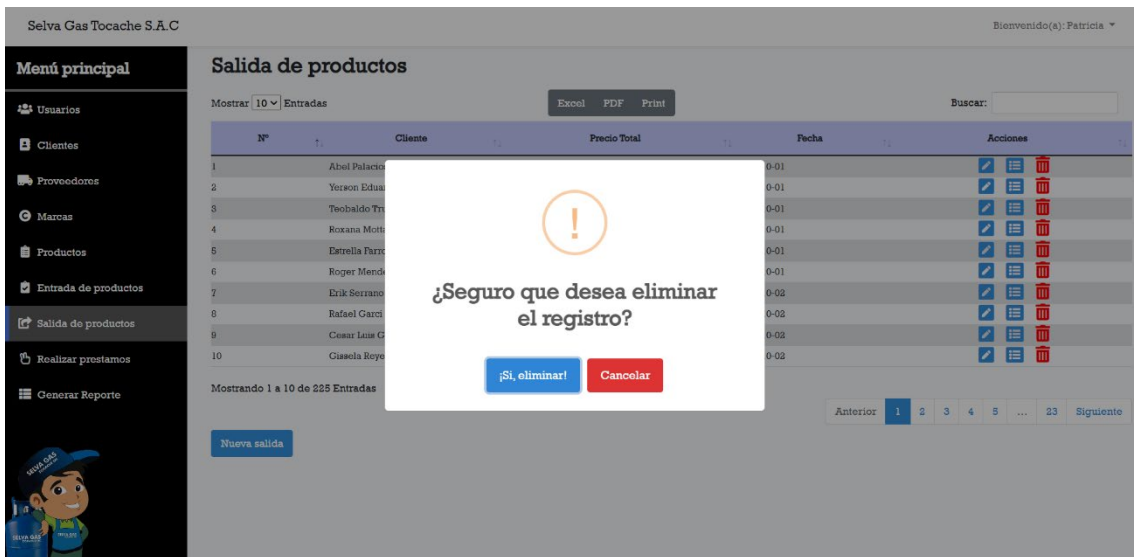


Figura N°116. Prototipo del módulo préstamos

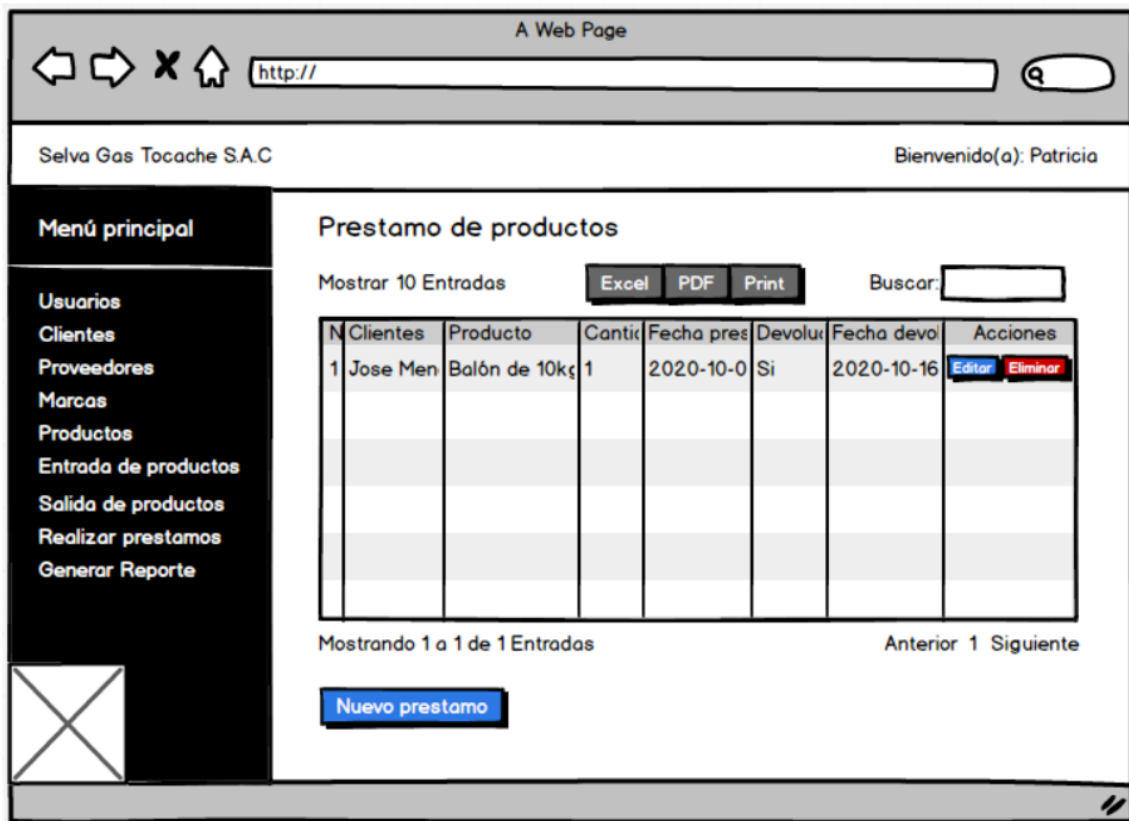


Figura N°117. Maquetación del módulo préstamos

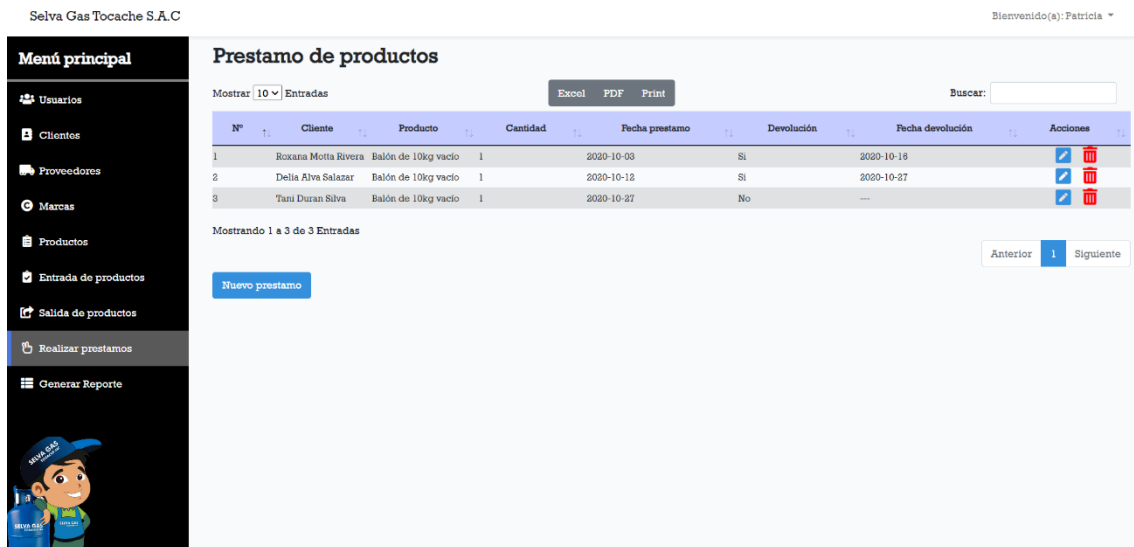


Figura N°118. Prototipo del módulo préstamos - registrar

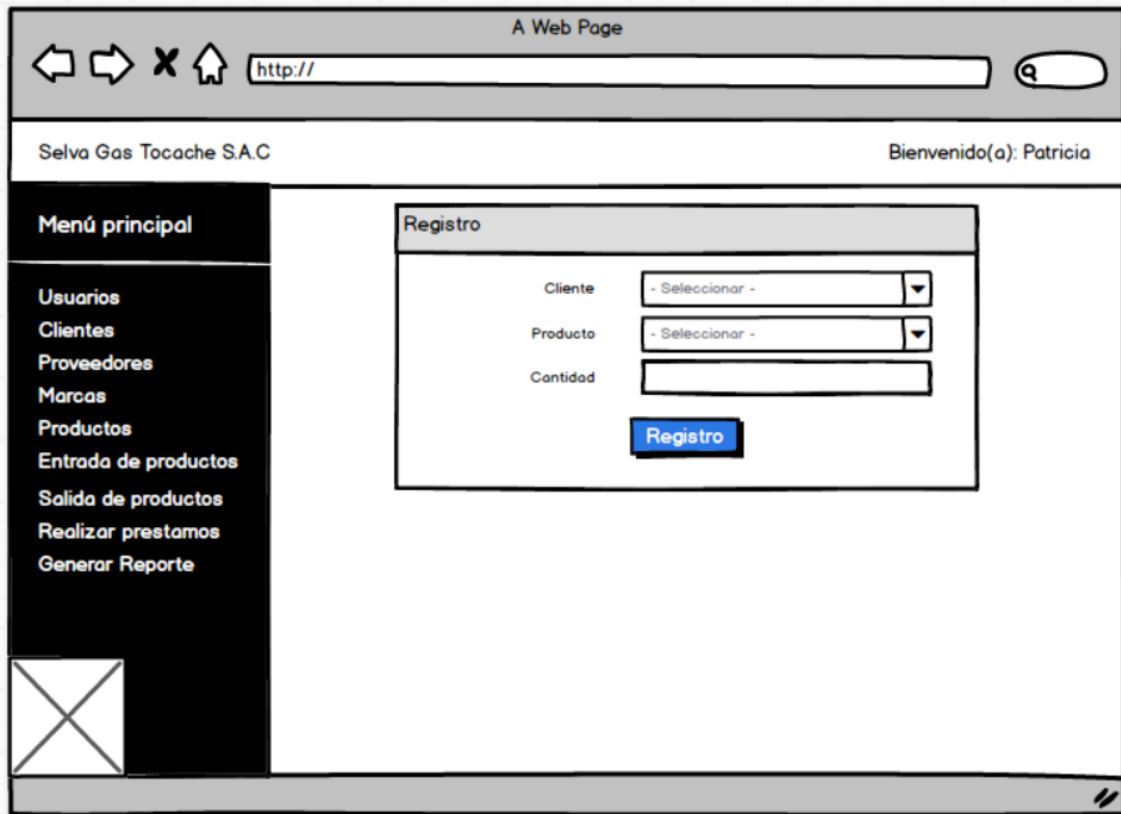


Figura N°119. Maquetación del módulo préstamos - registrar

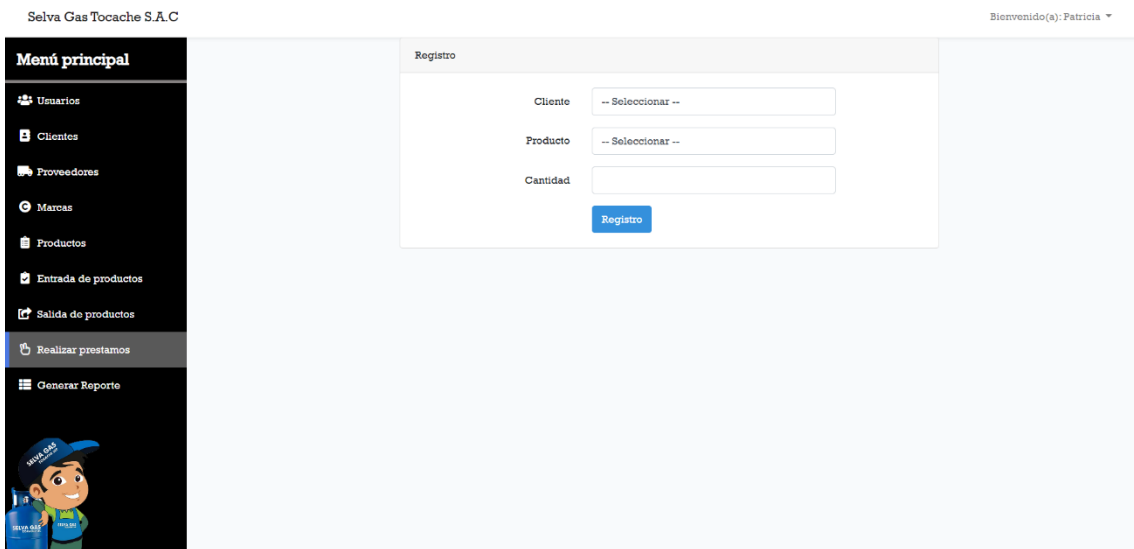


Figura N°120. Prototipo del módulo préstamos - editar

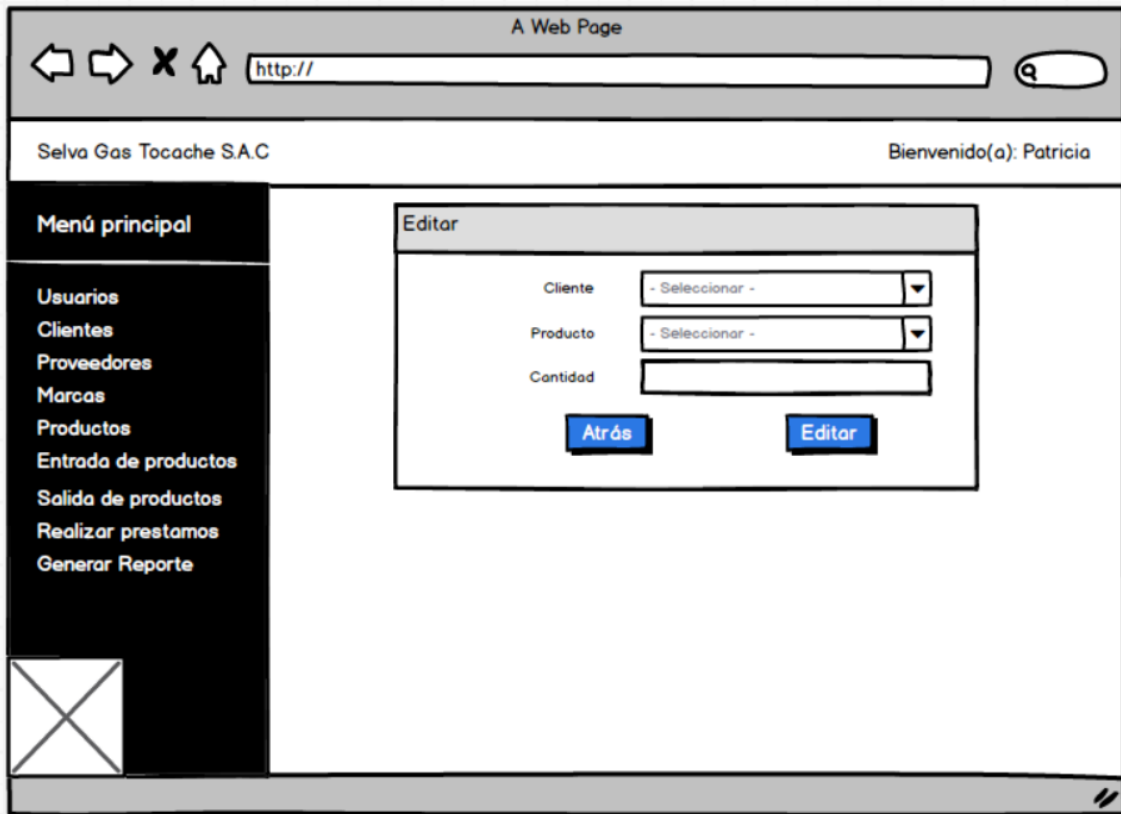


Figura N°121. Maquetación del módulo préstamos - editar

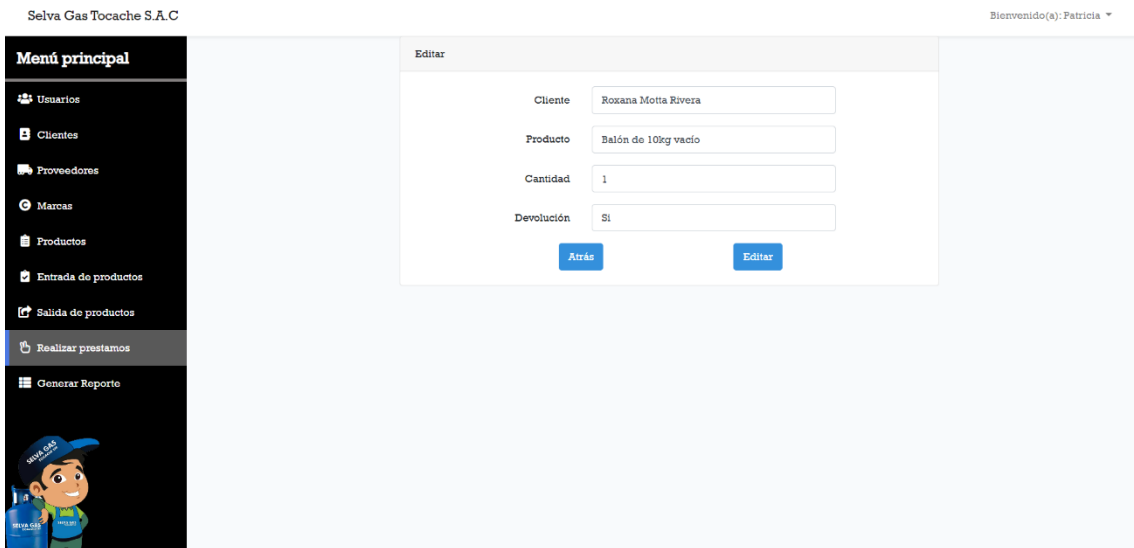


Figura N°122. Prototipo del módulo préstamos - eliminar



Figura N°123. Maquetación del módulo préstamos - eliminar

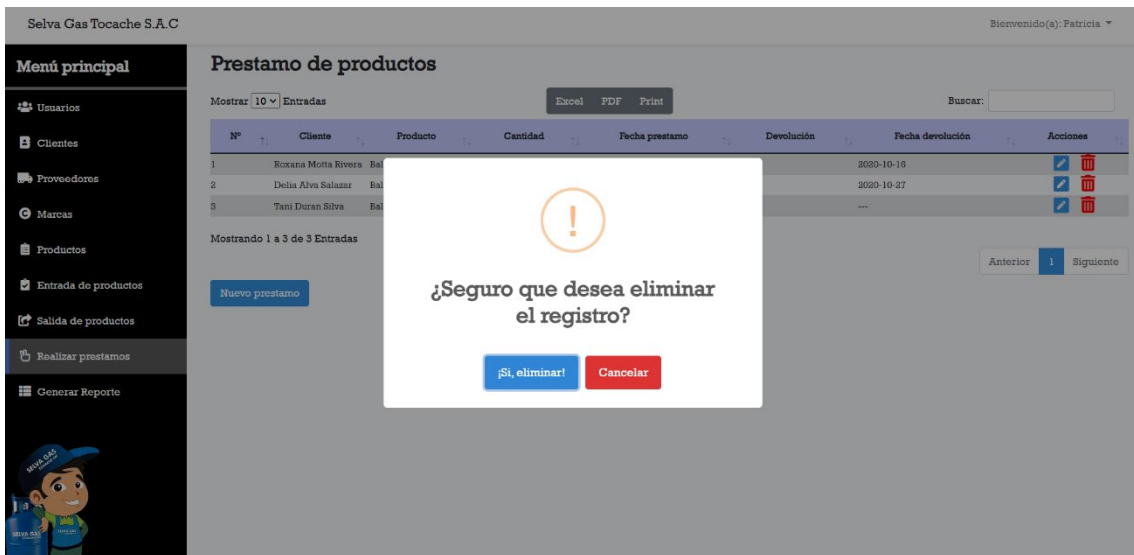




Figura N°124. Prototipo del módulo reporte

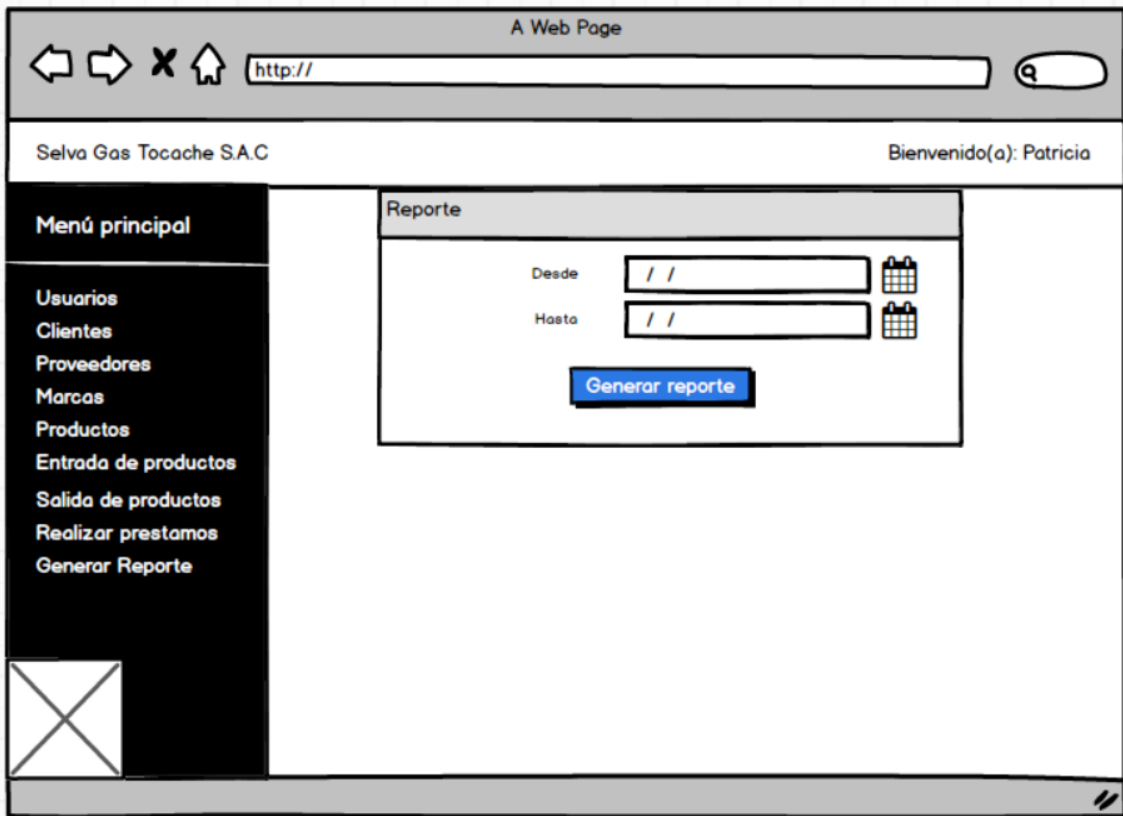


Figura N°125. Maquetación del módulo reporte

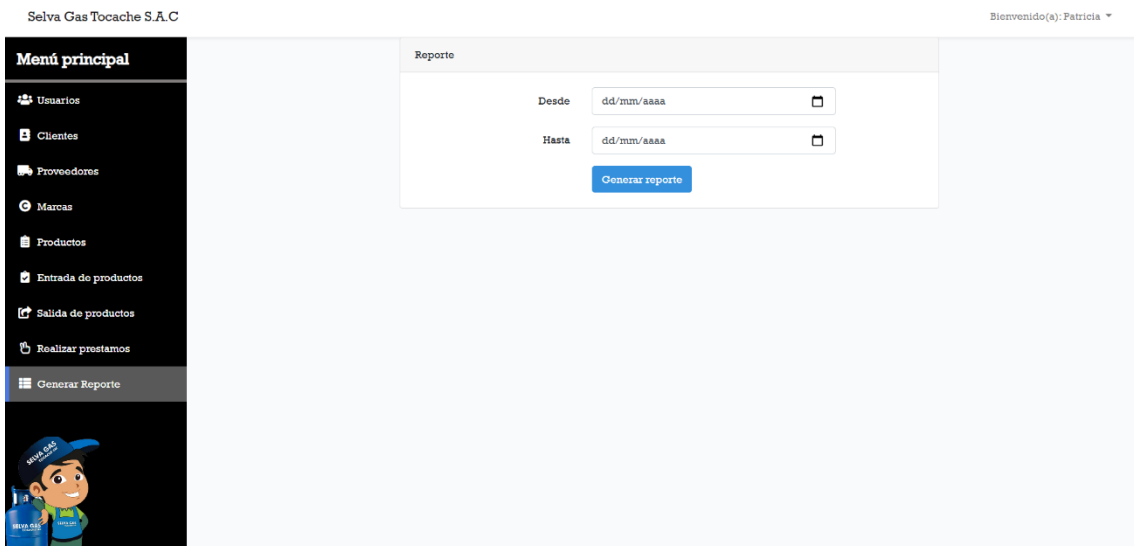


Figura N°126. Prototipo del módulo reporte – generar reporte

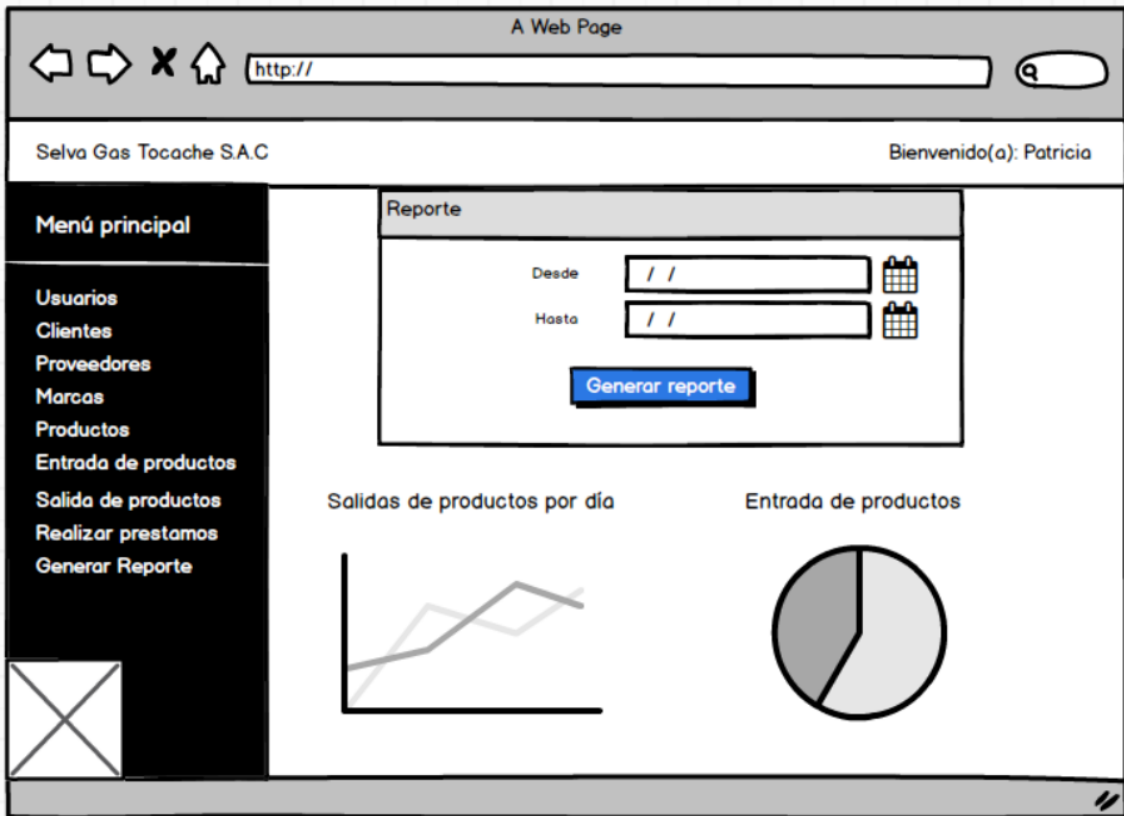
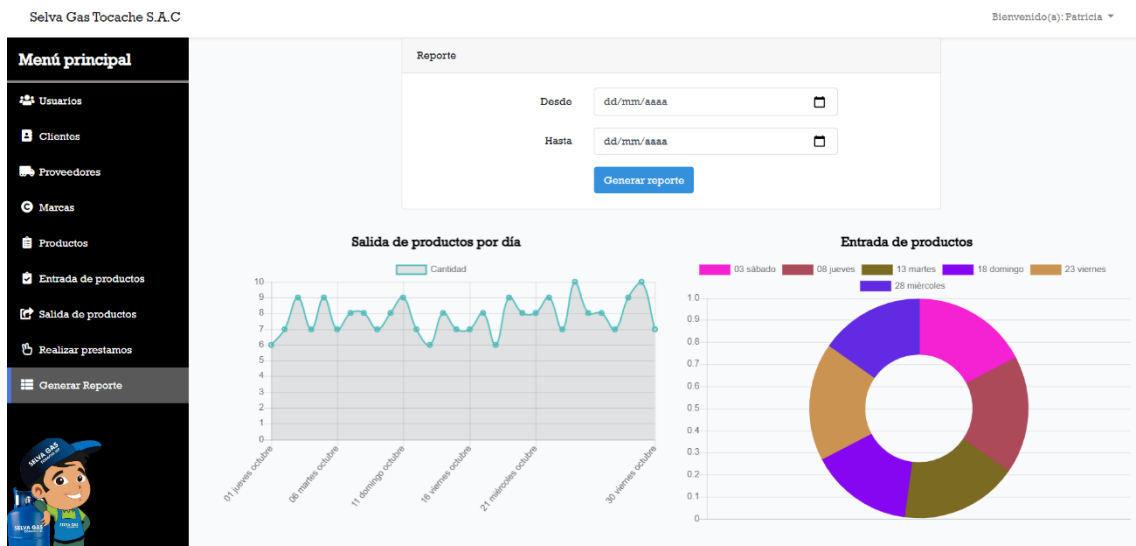


Figura N°127. Maquetación del módulo reporte – generar reporte



## Historia N° 3: Autenticación del sistema

Validación de los campos de la tabla usuarios con la BD.

Figura N°128. Código para validar las columnas de la tabla usuarios

```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8 use Illuminate\Notifications\Notifiable;
9 use Spatie\Permission\Traits\HasRoles;
10
11 class User extends Authenticatable
12 {
13     use Notifiable;
14     use HasRoles;
15     use SoftDeletes;
16     /**
17      * The attributes that are mass assignable.
18      *
19      * @var array
20      */
21     protected $table = 'users';
22
23     protected $dates = ['deleted_at'];
24
25     protected $fillable = [
26         'role_id', 'name', 'apellido_paterno', 'apellido_materno', 'email', 'password', 'direccion', 'telefono'
27     ];
28
29     /**
30      * The attributes that should be hidden for arrays.
31      *
32      * @var array
33      */
34     protected $hidden = [
35         'password', 'remember_token',
36     ];
37 }
```

Figura N°129. Código para listar los registros la tabla usuarios

```
24 public function datos_usuario_rol(){
25     $usuarios = DB::table('users as u')
26         ->select('u.id', 'u.name', 'u.apellido_paterno', 'u.apellido_materno', 'u.direccion', 'u.telefono', 'u.email',
27             'u.role_id', 'r.name as role_name')
28         ->join('roles as r', 'u.role_id', '=', 'r.id')
29         ->whereNull('u.deleted_at')
30         ->get();
31     return $usuarios;
32 }
33
34 public function index()
35 {
36     $usuarios = self::datos_usuario_rol();
37     return view('usuario.index', compact('usuarios'));
38 }
39 }
```

Figura N°130. Código para realizar un registro en la tabla usuarios

```

57 public function store(Request $request)
58 {
59     $request->validate([
60         'name' => ['required', 'string', 'max:255'],
61         'apellido_paterno' => ['required', 'string', 'max:255'],
62         'apellido_materno' => ['required', 'string', 'max:255'],
63         'direccion' => ['required', 'string', 'max:255'],
64         'role_id' => ['required'],
65         'password' => ['required', 'string', 'min:8', 'confirmed'],
66     ]);
67
68     if($request->telefono != null){
69         $request->validate([
70             'telefono' => ['bail', 'required', 'numeric', 'min:1000000', 'max:999999999'],
71         ]);
72     }
73
74     if($request->email != null){
75         $request->validate([
76             'email' => ['required', 'string', 'email', 'max:255'],
77         ]);
78     }
79
80     if($request['role_id'] == 1){
81
82         $user = User::create([
83             'name' => $request['name'],
84             'apellido_paterno' => $request['apellido_paterno'],
85             'apellido_materno' => $request['apellido_materno'],
86             'email' => $request['email'],
87             'direccion' => $request['direccion'],
88             'telefono' => $request['telefono'],
89             'role_id' => $request['role_id'],
90             'password' => Hash::make($request['password']),
91         ]);
92
93         $user->assignRole('administrador');
94
95         return redirect('/usuario')->with('success', 'Usuario registrado');
96     }else{
97
98         $user = User::create([
99             'name' => $request['name'],
100             'apellido_paterno' => $request['apellido_paterno'],
101             'apellido_materno' => $request['apellido_materno'],
102             'email' => $request['email'],
103             'direccion' => $request['direccion'],
104             'telefono' => $request['telefono'],
105             'role_id' => $request['role_id'],
106             'password' => Hash::make($request['password']),
107         ]);
108
109         $user->assignRole('usuario');
110
111         return redirect('/usuario')->with('success', 'Usuario registrado');

```

Figura N°131. Código para editar un registro en la tabla usuarios

```

151 public function update(Request $request, $id)
152 {
153     $request->validate([
154         'name' => ['required', 'string', 'max:255'],
155         'apellido_paterno' => ['required', 'string', 'max:255'],
156         'apellido_materno' => ['required', 'string', 'max:255'],
157         'direccion' => ['required', 'string', 'max:255'],
158         'role_id' => ['required'],
159         'password' => ['required', 'string', 'min:8', 'confirmed'],
160     ]);
161
162     if($request->telefono != null){
163         $request->validate([
164             'telefono' => ['bail', 'required', 'numeric', 'min:1000000', 'max:999999999'],
165         ]);
166     }
167
168     if($request->email != null){
169         $request->validate([
170             'email' => ['required', 'string', 'email', 'max:255'],
171         ]);
172     }
173
174     $usuario_update = User::findOrFail($id);
175     $usuario_update->name = $request->get('name');
176     $usuario_update->apellido_paterno = $request->get('apellido_paterno');
177     $usuario_update->apellido_materno = $request->get('apellido_materno');
178     $usuario_update->direccion = $request->get('direccion');
179     $usuario_update->telefono = $request->get('telefono');
180     $usuario_update->role_id = $request->get('role_id');
181     $usuario_update->email = $request->get('email');
182     $usuario_update->password = bcrypt($request->get('password'));
183
184     $usuario_update->save();
185
186     DB::table('model_has_roles')->where('model_id',$id)->update(['role_id' => $usuario_update->role_id]);
187
188     return redirect('/usuario')->with('success', 'Usuario actualizado');
189 }

```

Figura N°132. Código para eliminar un registro en la tabla usuarios

```
197 public function destroy($id)
198 {
199     $usuario_delete = User::find($id);
200     $usuario_delete->delete();
201
202     return redirect('/usuario')->with('success', 'Usuario eliminado');
203 }
204 }
205 }
```

Integración del código a la interfaz del login.

Figura N°133. Integración del código a la interfaz del login

```
1 @extends('layouts.app')
2
3 @section('main')
4 <p class="titulo_login">Selva Gas Tocache S.A.C</p>
5 <div class="row justify-content-center">
6     <div class="col-md-6">
7         <div class="card">
8             <div class="card-header text-center"><h2 class="m-0">{{ __('Acceso') }}</h2></div>
9             @if ($errors->any())
10                 <div class="alert alert-danger">
11                     <ul>
12                         @foreach ($errors->all() as $error)
13                             <li>{{ $error }}</li>
14                         @endforeach
15                     </ul>
16                 </div>
17             @endif
18             <div class="card-body">
19                 <div class="icono_user text-center">
20                     <i class="fas fa-user-circle"></i>
21                 </div>
22                 <form method="POST" action="{{ route('login') }}">
23                     @csrf
24
25                     <div class="form-group row">
26                         <div class="col-md-12">
27                             <div class="input-group">
28                                 <div class="input-group-text"><i class="fas fa-user"></i></div>
29                                 <input id="email" type="text" class="form-control @error('email') is-invalid @enderror" name="email" value="{{ old('email') }}"
30                                     * required autocomplete="email" autofocus placeholder="Username">
31                             </div>
32                             @error('email')
33                                 <span class="invalid-feedback" role="alert">
34                                     <strong>{{ $message }}</strong>
35                                 </span>
36                             @enderror
37                         </div>
38                     </div>
39                     <div class="form-group row">
40                         <div class="col-md-12">
41                             <div class="input-group">
42                                 <div class="input-group-text"><i class="fas fa-lock"></i></div>
43                                 <input id="password" type="password" class="form-control @error('password') is-invalid @enderror" name="password" value="{{
44                                     old('password') }}" required autocomplete="password" autofocus placeholder="Password">
45                             </div>
46                             @error('password')
47                                 <span class="invalid-feedback" role="alert">
48                                     <strong>{{ $message }}</strong>
49                                 </span>
50                             @enderror
51                         </div>
52                     </div>
```

### Acta de entrega del Sprint 1

Siendo las 19:00 horas del día 10 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Lesly Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura a los requerimientos realizados y encargados del Sprint 1, asimismo muestra los prototipos, interfaces y el código realizado hasta el momento.

Verificadas las explicaciones y el sustento presentado por el señor Romero Rodríguez, Billy Jerson para la aprobación del Sprint 1, se decide aprobar por parte de la gerente general Estupiñan Palma, Patricia el término del Sprint 1, del proyecto "Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C".



\_\_\_\_\_  
Firma

## Sprint 2

A continuación, se mostrarán los requerimientos funcionales que se realizarán para el sprint 2.

Tabla N°35: *Sprint 2*

#	Requerimientos Funcionales	N° Historia	Nombre Historia	Prioridad	Tiempo
1	Cada rol de usuario ya debe estar predeterminado, para agilizar la creación de nuevas credenciales.	H4	Establecer permisos de usuarios	Alta	2
2	Permita registrar, editar y eliminar registros de clientes.	H5	Mantenimiento de los clientes	Media	3
	Permita exportar los datos.				
3	Permita registrar, editar y eliminar registros de proveedores.	H6	Mantenimiento de los proveedores	Media	3
	Permita exportar los datos.				
4	Permita registrar, editar y eliminar registros de marcas.	H7	Mantenimiento de las marcas	Media	2
	Permita exportar los datos.				
Total					10

## Planificación del Sprint 2

Siendo las 8:30 horas del día 11 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Lesly Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura de los requerimientos del sistema.

Analizada la lista de requerimientos por el señor Romero Rodríguez Billy Jerson solventa algunas dudas sobre los requerimientos y se compromete a cumplir con los requisitos establecidos por el Sprint 2.

Los presentes imparten su aprobación a los requerimientos de acuerdo a lo establecido en la planificación del Sprint 2 indicando que la fecha de entrega de este Sprint sería el día 20 de abril de 2021.



Firma



## Cronograma de tareas del Sprint 2

Tabla N°36: Cronograma del Sprint 2

<b>Sprint 2</b>	<b>Requerimiento F.</b>	10 días	11/04/2021	20/04/2021
<b>Historia N°4: Establecer permisos de usuario</b>	<b>RF6</b>	2 días	11/04/2021	12/04/2021
Creación de los permisos para cada rol.		1 día	11/04/2021	11/04/2021
Codificación predefinidos por el sistema.		1 día	12/04/2021	12/04/2021
<b>Historia N°5: Mantenimiento de los clientes</b>	<b>RF13, RF14</b>	3 días	13/04/2021	15/04/2021
Elaboración de procedimientos para el CRUD de datos.		1 día	13/04/2021	13/04/2021
Integración del código a la interfaz del módulo.		1 día	14/04/2021	14/04/2021
Comprobación de los campos de texto para el CRUD.		1 día	15/04/2021	15/04/2021
<b>Historia N°6: Mantenimiento de los proveedores</b>	<b>RF15, RF16</b>	3 días	16/04/2021	18/04/2021
Elaboración de procedimientos para el CRUD de datos.		1 día	16/04/2021	16/04/2021
Integración del código a la interfaz del módulo.		1 día	17/04/2021	17/04/2021
Comprobación de los campos de texto para el CRUD.		1 día	18/04/2021	18/04/2021
<b>Historia N°7: Mantenimiento de las marcas</b>	<b>RF17, RF18</b>	2 días	19/04/2021	20/04/2021
Elaboración de procedimientos para el CRUD de datos.		1 día	19/04/2021	19/04/2021
Integración del código y comprobación de los campos con la interfaz		1 día	20/04/2021	20/04/2021

En la tabla anterior se muestra el cronograma del Sprint 2, en donde se muestra las fechas en las que se realizarán los requerimientos funcionales con respecto a cada una de sus historias de usuario.

## Historia N° 4: Creación de la base de datos

Creación de los permisos para cada rol.

Figura N°134. Creación de permisos para cada rol

```
25 //Index
26 Permission::create(['name' => 'cliente.index']);
27 Permission::create(['name' => 'detalle_salida.index']);
28 Permission::create(['name' => 'marca.index']);
29 Permission::create(['name' => 'menu.index']);
30 Permission::create(['name' => 'pedido.index']);
31 Permission::create(['name' => 'prestamo.index']);
32 Permission::create(['name' => 'producto.index']);
33 Permission::create(['name' => 'proveedor.index']);
34 Permission::create(['name' => 'reporte.index']);
35 Permission::create(['name' => 'salida.index']);
36 Permission::create(['name' => 'usuario.index']);
37
38 //Create
39 Permission::create(['name' => 'cliente.create']);
40 Permission::create(['name' => 'detalle_salida.create']);
41 Permission::create(['name' => 'marca.create']);
42 Permission::create(['name' => 'pedido.create']);
43 Permission::create(['name' => 'prestamo.create']);
44 Permission::create(['name' => 'producto.create']);
45 Permission::create(['name' => 'proveedor.create']);
46 Permission::create(['name' => 'salida.create']);
47 Permission::create(['name' => 'usuario.create']);
48
49 //Edit
50 Permission::create(['name' => 'cliente.edit']);
51 Permission::create(['name' => 'detalle_salida.edit']);
52 Permission::create(['name' => 'marca.edit']);
53 Permission::create(['name' => 'pedido.edit']);
54 Permission::create(['name' => 'prestamo.edit']);
55 Permission::create(['name' => 'producto.edit']);
56 Permission::create(['name' => 'proveedor.edit']);
57 Permission::create(['name' => 'salida.edit']);
58 Permission::create(['name' => 'usuario.edit']);
59
60 //Update
61 Permission::create(['name' => 'cliente.update']);
62 Permission::create(['name' => 'detalle_salida.update']);
63 Permission::create(['name' => 'marca.update']);
64 Permission::create(['name' => 'pedido.update']);
65 Permission::create(['name' => 'prestamo.update']);
66 Permission::create(['name' => 'producto.update']);
67 Permission::create(['name' => 'proveedor.update']);
68 Permission::create(['name' => 'salida.update']);
69 Permission::create(['name' => 'usuario.update']);
70
```

Codificación predefinidos por el sistema.

Figura N°135. Permisos predefinidos - administrador

```
82 $admin = Role::create(['name' => 'administrador']);
83
84 $admin->givePermissionTo([
85     'cliente.index',
86     'detalle_salida.index',
87     'marca.index',
88     'menu.index',
89     'pedido.index',
90     'prestamo.index',
91     'producto.index',
92     'proveedor.index',
93     'reporte.index',
94     'salida.index',
95     'usuario.index',
96     'cliente.create',
97     'detalle_salida.create',
98     'marca.create',
99     'pedido.create',
100     'prestamo.create',
101     'producto.create',
102     'proveedor.create',
103     'salida.create',
104     'usuario.create',
105     'cliente.edit',
106     'detalle_salida.edit',
107     'marca.edit',
108     'pedido.edit',
109     'prestamo.edit',
110     'producto.edit',
111     'proveedor.edit',
112     'salida.edit',
113     'usuario.edit',
114     'cliente.update',
115     'detalle_salida.update',
116     'marca.update',
117     'pedido.update',
118     'prestamo.update',
119     'producto.update',
120     'proveedor.update',
121     'salida.update',
122     'usuario.update',
123     'cliente.destroy',
124     'detalle_salida.destroy',
125     'marca.destroy',
126     'pedido.destroy',
127     'prestamo.destroy',
128     'producto.destroy',
129     'proveedor.destroy',
130     'salida.destroy',
131     'usuario.destroy',
132 ]);
```

Figura N°136. Permisos predefinidos - usuario

```
134     $guest = Role::create(['name' => 'usuario']);
135
136     $guest->givePermissionTo([
137         'cliente.index',
138         'cliente.create',
139         'cliente.edit',
140         'cliente.update',
141         'cliente.destroy',
142         'salida.index',
143         'salida.create',
144         'salida.edit',
145         'salida.update',
146         'salida.destroy',
147         'detalle_salida.index',
148         'detalle_salida.create',
149         'detalle_salida.edit',
150         'detalle_salida.update',
151         'detalle_salida.destroy',
152         'prestamo.index',
153         'prestamo.create',
154         'prestamo.edit',
155         'prestamo.update',
156         'prestamo.destroy',
157         'pedido.index',
158         'pedido.create',
159         'pedido.edit',
160         'pedido.update',
161         'pedido.destroy',
162         'producto.index',
163         'producto.create',
164         'producto.edit',
165         'producto.update',
166         'producto.destroy',
167     ]);
```

## Historia N°5: Mantenimiento de los clientes

Elaboración de procedimientos para el CRUD de datos.

Figura N°137. Cliente - Registrar

```
45     public function store(Request $request)
46     {
47         $request->validate([
48             'nombre' => ['required', 'regex:/^[^\pL\s\~]+$/u'],
49             'apellido_paterno' => ['required', 'regex:/^[^\pL\s\~]+$/u'],
50             'apellido_materno' => ['required', 'regex:/^[^\pL\s\~]+$/u'],
51             'direccion' => ['required', 'string', 'max:255'],
52         ]);
53
54         $user= auth()->user()->id;
55
56         if($request->telefono != null){
57             $request->validate([
58                 'telefono' => ['bail', 'required', 'numeric', 'min:1000000', 'max:999999999'],
59             ]);
60         }
61
62         if($request->email != null){
63             $request->validate([
64                 'email' => ['required', 'string', 'email', 'max:255'],
65             ]);
66         }
67
68         $clientes = new ClienteModel([
69             'user_id' => $user,
70             'nombre' => $request->get('nombre'),
71             'apellido_paterno' => $request->get('apellido_paterno'),
72             'apellido_materno' => $request->get('apellido_materno'),
73             'direccion' => $request->get('direccion'),
74             'referencia' => $request->get('referencia'),
75             'telefono' => $request->get('telefono'),
76             'email' => $request->get('email'),
77         ]);
78
79         $clientes->save();
80
81         return redirect('/cliente')->with('success', 'Cliente registrado');
82     }
```

Figura N°138. Cliente - Editar

```

115 public function update(Request $request, $id)
116 {
117     $request->validate([
118         'nombre' => ['required', 'regex:/^[a-zA-Z\s-]+$/u'],
119         'apellido_paterno' => ['required', 'regex:/^[a-zA-Z\s-]+$/u'],
120         'apellido_materno' => ['required', 'regex:/^[a-zA-Z\s-]+$/u'],
121         'direccion' => ['required', 'string', 'max:255'],
122     ]);
123
124     if($request->telefono != null){
125         $request->validate([
126             'telefono' => ['bail', 'required', 'numeric', 'min:1000000', 'max:999999999'],
127         ]);
128     }
129
130     if($request->email != null){
131         $request->validate([
132             'email' => ['required', 'string', 'email', 'max:255'],
133         ]);
134     }
135
136     $cliente_update = ClienteModel::find($id);
137     $cliente_update->nombre = $request->get('nombre');
138     $cliente_update->apellido_paterno = $request->get('apellido_paterno');
139     $cliente_update->apellido_materno = $request->get('apellido_materno');
140     $cliente_update->direccion = $request->get('direccion');
141     $cliente_update->referencia = $request->get('referencia');
142     $cliente_update->telefono = $request->get('telefono');
143     $cliente_update->email = $request->get('email');
144     $cliente_update->save();
145
146     return redirect('/cliente')->with('success', 'Cliente actualizado');
147 }

```

Figura N°139. Cliente - Eliminar

```

155 public function destroy($id)
156 {
157     $cliente_eliminar = ClienteModel::find($id);
158     $cliente_eliminar->delete();
159
160     return redirect('/cliente')->with('success', 'Cliente eliminado');
161 }

```

Integración del código a la interfaz del módulo.

Figura N°140. Cliente - Interfaz

```

1 @extends('inicio')
2
3 @section('main')
4 <div class="loading"></div>
5 <div class="row">
6 <div class="col-sm-12">
7 <table id="tabla_general" class="table table-striped">
8 <thead>
9 <tr>
10 <th></th>
11 <th>Nombres y Apellidos</th>
12 <th>Direccion</th>
13 <th>telefono</th>
14 <th>Acciones</th>
15 </tr>
16 </thead>
17 <tbody>
18 <tr>
19 <td><div class="loading"></div></td>
20 <td><div class="loading"></div></td>
21 <td><div class="loading"></div></td>
22 <td><div class="loading"></div></td>
23 <td><div class="loading"></div></td>
24 <td><div class="loading"></div></td>
25 <td><div class="loading"></div></td>
26 <td><div class="loading"></div></td>
27 <td><div class="loading"></div></td>
28 <td><div class="loading"></div></td>
29 <td><div class="loading"></div></td>
30 <td><div class="loading"></div></td>
31 <td><div class="loading"></div></td>
32 <td><div class="loading"></div></td>
33 <td><div class="loading"></div></td>
34 <td><div class="loading"></div></td>
35 <td><div class="loading"></div></td>
36 <td><div class="loading"></div></td>
37 <td><div class="loading"></div></td>
38 <td><div class="loading"></div></td>
39 <td><div class="loading"></div></td>
40 <td><div class="loading"></div></td>
41 <td><div class="loading"></div></td>
42 <td><div class="loading"></div></td>
43 <td><div class="loading"></div></td>
44 <td><div class="loading"></div></td>
45 <td><div class="loading"></div></td>
46 <td><div class="loading"></div></td>
47 <td><div class="loading"></div></td>
48 <td><div class="loading"></div></td>
49 <td><div class="loading"></div></td>
50 <td><div class="loading"></div></td>

```

## Comprobación de los campos de texto para el CRUD.

Figura N°141. Cliente – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4
5 <div class="container">
6   <div class="row justify-content-center">
7     <div class="col-md-8">
8       <div class="card">
9         <div class="card-header">{{ __('Register') }}</div>
10
11         <div class="card-body">
12           <form method="POST" action="{{ route('cliente.store') }}">
13             @csrf
14
15             <div class="form-group row">
16               <label for="nombre" class="col-md-4 col-form-label text-md-right">{{ __('Nombre') }}</label>
17
18               <div class="col-md-6">
19                 <input id="nombre" type="text" class="form-control @error('nombre') is-invalid @enderror" name="nombre" value="{{ old('nombre') }}" required autocomplete="nombre" autofocus>
20
21                 @error('nombre')
22                   <span class="invalid-feedback" role="alert">
23                     <strong>{{ $message }}</strong>
24                   </span>
25                 @enderror
26               </div>
27             </div>
28
29             <div class="form-group row">
30               <label for="apellido_paterno" class="col-md-4 col-form-label text-md-right">{{ __('Apellido Paterno') }}</label>
31
32               <div class="col-md-6">
33                 <input id="apellido_paterno" type="text" class="form-control @error('apellido_paterno') is-invalid @enderror" name="apellido_paterno" value="{{ old('apellido_paterno') }}" required autocomplete="apellido_paterno" autofocus>
34
35                 @error('apellido_paterno')
36                   <span class="invalid-feedback" role="alert">
37                     <strong>{{ $message }}</strong>
38                   </span>
39                 @enderror
40               </div>
41             </div>
42
43             <div class="form-group row">
44               <label for="apellido_materno" class="col-md-4 col-form-label text-md-right">{{ __('Apellido Materno') }}</label>
45
46               <div class="col-md-6">
47                 <input id="apellido_materno" type="text" class="form-control @error('apellido_materno') is-invalid @enderror" name="apellido_materno" value="{{ old('apellido_materno') }}" required autocomplete="apellido_materno" autofocus>
48
49                 @error('apellido_materno')
50                   <span class="invalid-feedback" role="alert">
51                     <strong>{{ $message }}</strong>
52                   </span>
53                 @enderror
54               </div>
55             </div>
56           </form>
57         </div>
58       </div>
59     </div>
60   </div>
61 </div>
```

## Historia N°6: Mantenimiento de los proveedores

Elaboración de procedimientos para el CRUD de datos.

Figura N°142. Proveedor - Registrar

```
39 public function store(Request $request)
40 {
41     $request->validate([
42         'nombre' => ['required', 'regex:/^[a-zA-Z\s-]+$/u', 'max:255'],
43         'ruc' => ['required', 'numeric', 'min:1000000000', 'max:9999999999'],
44         'direccion' => ['required', 'string', 'max:255'],
45     ]);
46
47     if($request->telefono != null){
48         $request->validate([
49             'telefono' => ['bail', 'required', 'numeric', 'min:100000', 'max:999999999'],
50         ]);
51     }
52
53     if($request->email != null){
54         $request->validate([
55             'email' => ['required', 'string', 'email', 'max:255'],
56         ]);
57     }
58
59     $proveedores = new ProveedorModel([
60         'nombre' => $request->get('nombre'),
61         'ruc' => $request->get('ruc'),
62         'direccion' => $request->get('direccion'),
63         'telefono' => $request->get('telefono'),
64         'email' => $request->get('email'),
65     ]);
66
67     $proveedores->save();
68
69     return redirect('/proveedor')->with('success', 'Proveedor registrado');
70 }
```

Figura N°143. Proveedor - Editar

```
103 public function update(Request $request, $id)
104 {
105     $request->validate([
106         'nombre' => ['required', 'regex:/^[a-zA-Z\s-]+$/u', 'max:255'],
107         'ruc' => ['required', 'numeric', 'min:1000000000', 'max:9999999999'],
108         'direccion' => ['required', 'string', 'max:255'],
109     ]);
110
111     if($request->telefono != null){
112         $request->validate([
113             'telefono' => ['bail', 'required', 'numeric', 'min:100000', 'max:999999999'],
114         ]);
115     }
116
117     if($request->email != null){
118         $request->validate([
119             'email' => ['required', 'string', 'email', 'max:255'],
120         ]);
121     }
122
123     $proveedor_update = ProveedorModel::find($id);
124     $proveedor_update->nombre = $request->get('nombre');
125     $proveedor_update->ruc = $request->get('ruc');
126     $proveedor_update->direccion = $request->get('direccion');
127     $proveedor_update->telefono = $request->get('telefono');
128     $proveedor_update->email = $request->get('email');
129     $proveedor_update->save();
130
131     return redirect('/proveedor')->with('success', 'Proveedor actualizado!');
132 }
```

Figura N°144. Proveedor - Eliminar

```
140 public function destroy($id)
141 {
142     $proveedor_eliminar = ProveedorModel::find($id);
143     $proveedor_eliminar->delete();
144
145     return redirect('/proveedor')->with('success', 'Proveedor eliminado!');
146 }
```

Integración del código a la interfaz del módulo.

Figura N°145. Proveedor - Interfaz

```
1 @extends('inicio')
2
3 @section('main')
4 <div class="row">
5 <div class="col-sm-12">
6 <table id="tabla_general" class="table table-striped">
7 <thead>
8 <tr>
9 <th>N°</th>
10 <th>Nombre</th>
11 <th>Dirección</th>
12 <th>Teléfono</th>
13 <th>Acciones</th>
14 </tr>
15 </thead>
16 <tbody>
17 <tr>
18 <td>?php $i-1 ?>
19 <td>?php $b-1 ?>
20 @foreach($proveedores as $prov)
21 <td><span>{{ $i++ }}</span></td>
22 <td><span>{{ $prov->nombre }}</span></td>
23 <td><span>{{ $prov->direccion }}</span></td>
24 <td><span>{{ $prov->telefono }}</span></td>
25 <td><div class="botones">
26 <a class="accion_editar" data-bs-toggle="tooltip" title="Editar" href="{{ route('proveedor.edit', $prov->id)}}"><i class="fas fa-pen-square"></i></a>
27 <button class="accion_mostrar" data-bs-toggle="modal" data-bs-target="#exampleModal{{ $b++ }}"><i class="far fa-address-book"></i></button>
28 <form id="{{ $prov->id }}" action="{{ route('proveedor.destroy', $prov->id)}}" method="post">
29 @csrf
30 <input type="button" value="Eliminar" class="accion_eliminar" onclick="eliminarRegistro(event)"></input>
31 </form>
32 </td>
33 </tr>
34 @endforeach
35 </tbody>
36 </table>
37 <div style="margin: 5px 0px 0px 0px;" href="{{ route('proveedor.create')}}" class="btn btn-primary">Nuevo proveedor</div>
38
39 @if (session('success'))
40 <div class="alert alert-success">
41 {{ session('success') }}
42 </div>
43 @endif
```

## Comprobación de los campos de texto para el CRUD.

Figura N°146. Proveedor – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4
5 <div class="container">
6   <div class="row justify-content-center">
7     <div class="col-md-8">
8       <div class="card">
9         <div class="card-header">{{ __('Register') }}</div>
10
11         <div class="card-body">
12           <form method="POST" action="{{ route('proveedor.store') }}">
13             @csrf
14
15             <div class="form-group row">
16               <label for="nombre" class="col-md-4 col-form-label text-md-right">{{ __('Nombre') }}</label>
17
18               <div class="col-md-6">
19                 <input id="nombre" type="text" class="form-control @error('nombre') is-invalid @enderror" name="nombre" value="{{ old('nombre') }}" required autocomplete="nombre" autofocus>
20
21                 @error('nombre')
22                   <span class="invalid-feedback" role="alert">
23                     <strong>{{ $message }}</strong>
24                   </span>
25                 @enderror
26               </div>
27             </div>
28
29             <div class="form-group row">
30               <label for="ruc" class="col-md-4 col-form-label text-md-right">{{ __('RUC') }}</label>
31
32               <div class="col-md-6">
33                 <input id="ruc" type="text" class="form-control @error('ruc') is-invalid @enderror" name="ruc" value="{{ old('ruc') }}" required autocomplete="ruc" autofocus>
34
35                 @error('ruc')
36                   <span class="invalid-feedback" role="alert">
37                     <strong>{{ $message }}</strong>
38                   </span>
39                 @enderror
40               </div>
41             </div>
42
43             <div class="form-group row">
44               <label for="direccion" class="col-md-4 col-form-label text-md-right">{{ __('Dirección') }}</label>
45
46               <div class="col-md-6">
47                 <input id="direccion" type="text" class="form-control @error('direccion') is-invalid @enderror" name="direccion" value="{{ old('direccion') }}" required autocomplete="direccion" autofocus>
48
49                 @error('direccion')
50                   <span class="invalid-feedback" role="alert">
51                     <strong>{{ $message }}</strong>
52                   </span>
53                 @enderror
54               </div>
55             </div>
56           </form>
57         </div>
58       </div>
59     </div>
60   </div>
61 </div>
```

## Historia N°7: Mantenimiento de las marcas

Elaboración de procedimientos para el CRUD de datos.

Figura N°147. Marca - Registrar

```
38 public function store(Request $request)
39 {
40     $request->validate([
41         'nombre' => ['required', 'string', 'max:255'],
42     ]);
43
44     $marcas = new MarcaModel([
45         'nombre' => $request->get('nombre'),
46     ]);
47
48     $marcas->save();
49
50     return redirect('/marca')->with('success', 'Marca registrada');
51 }
```

Figura N°148. Marca - Editar

```
84 public function update(Request $request, $id)
85 {
86     $request->validate([
87         'nombre' => ['required', 'string', 'max:255'],
88     ]);
89
90     $marca_update = MarcaModel::find($id);
91     $marca_update->nombre = $request->get('nombre');
92     $marca_update->save();
93
94     return redirect('/marca')->with('success', 'Marca actualizada!');
95 }
```

Figura N°149. Marca - Eliminar

```
103 public function destroy($id)
104 {
105     $marca_eliminar = MarcaModel::find($id);
106     $marca_eliminar->delete();
107
108     return redirect('/marca')->with('success', 'Marca eliminado!');
109 }
110 }
```

Integración del código y comprobación de los campos con la interfaz

Figura N°150. Marca - Interfaz

```
1 @extends('inicio')
2
3 @section('main')
4 <div class="row">
5 <div class="col-sm-12">
6 <table id="tabla_general" class="table table-striped">
7 <thead>
8 <tr>
9 <th>N°</th>
10 <th>Nombre</th>
11 <th>Acciones</th>
12 </tr>
13 </thead>
14 <tbody>
15 <tr>
16 <td colspan="3">
17 <div class="botones">
18 <div class="row">
19 <div class="col-sm-12">
20 <div class="col-sm-12">
21 <div class="col-sm-12">
22 <div class="col-sm-12">
23 <div class="col-sm-12">
24 <div class="col-sm-12">
25 <div class="col-sm-12">
26 <div class="col-sm-12">
27 <div class="col-sm-12">
28 <div class="col-sm-12">
29 <div class="col-sm-12">
30 <div class="col-sm-12">
31 <div class="col-sm-12">
32 <div class="col-sm-12">
33 <div class="col-sm-12">
34 <div class="col-sm-12">
35 <div class="col-sm-12">
36 <div class="col-sm-12">
37 <div class="col-sm-12">
38 <div class="col-sm-12">
39 <div class="col-sm-12">
40 <div class="col-sm-12">
41 <div class="col-sm-12">
42 <div class="col-sm-12">
43 <div class="col-sm-12">
44 <div class="col-sm-12">
45 @endsection
```

Figura N°151. Marca – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4 <div class="container">
5 <div class="row justify-content-center">
6 <div class="col-md-8">
7 <div class="card">
8 <div class="card-header">{{ __('Register') }}</div>
9 <div class="card-body">
10 <form method="POST" action="{{ route('marca.store') }}">
11 @csrf
12 <div class="form-group row">
13 <div class="col-md-4 col-form-label text-md-right">{{ __('Nombre') }}</div>
14 <div class="col-md-8">
15 <input id="nombre" type="text" class="form-control @error('nombre') is-invalid @enderror" name="nombre" value="{{ old('nombre') }}" required autocomplete="nombre" autofocus>
16 @error('nombre')
17 <span class="invalid-feedback" role="alert">
18 <strong>{{ $message }}</strong>
19 </span>
20 @enderror
21 </div>
22 </div>
23 <div class="form-group row mb-0">
24 <div class="col-md-6 offset-md-4">
25 <button type="submit" class="btn btn-primary">
26 <span>{{ __('Register') }}</span>
27 </button>
28 </div>
29 </div>
30 </div>
31 </div>
32 </div>
33 </div>
34 </div>
35 </div>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 @endsection
```



### Acta de entrega del Sprint 2

Siendo las 20:00 horas del día 20 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Lesly Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura a los requerimientos realizados y encargados del Sprint 2, asimismo muestra las interfaces y el código realizado hasta el momento.

Verificadas las explicaciones y el sustento presentado por el señor Romero Rodríguez, Billy Jerson para la aprobación del Sprint 2, se decide aprobar por parte de la gerente general Estupiñan Palma, Patricia el término del Sprint 2, del proyecto "Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C".



Firma

### Sprint 3

A continuación, se mostrarán los requerimientos funcionales que se realizarán para el sprint 3.

Tabla N°37: *Sprint 3*

#	Requerimientos Funcionales	N° Historia	Nombre Historia	Prioridad	Tiempo
1	Permita registrar, editar y eliminar registros de productos.	H8	Mantenimiento de los productos	Media	4
	Permita exportar los datos.				
2	Muestre de forma rápida si un producto cuenta con suficiente stock.	H9	Alerta de stock mínimo	Alta	2
	Cada rango de cantidad será mostrado de un color diferente, para su fácil interpretación.				
3	Permita registrar, editar y eliminar registros de las entradas de los productos.	H10	Mantenimiento de las entradas de productos	Media	4
	Permita exportar los datos.				
	Permita exportar los datos.				
Total					10

### Planificación del Sprint 3

Siendo las 9:30 horas del día 21 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Lesly Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura de los requerimientos del sistema.

Analizada la lista de requerimientos por el señor Romero Rodríguez Billy Jerson solventa algunas dudas sobre los requerimientos y se compromete a cumplir con los requisitos establecidos por el Sprint 3.

Los presentes imparten su aprobación a los requerimientos de acuerdo a lo establecido en la planificación del Sprint 3 indicando que la fecha de entrega de este Sprint sería el día 30 de abril de 2021.



Firma

### Cronograma de tareas del Sprint 3

Tabla N°38: Cronograma del Sprint 3

<b>Sprint 3</b>	<b>Requerimiento F.</b>	10 días	21/04/2021	30/04/2021
<b>Historia N°8: Mantenimiento de los productos</b>	<b>RF19, RF20</b>	4 días	21/04/2021	24/04/2021
Elaboración de procedimientos para el CRUD de datos.		1 día	21/04/2021	21/04/2021
Integración del código a la interfaz del módulo.		2 día	22/04/2021	23/04/2021
Comprobación de los campos de texto para el CRUD.		1 día	24/04/2021	24/04/2021
<b>Historia N°9: Alerta de stock mínimo</b>		<b>RF7, RF8</b>	2 días	25/04/2021
Elaboración del procedimiento para obtener los datos.	1 día		25/04/2021	25/04/2021
Integración del código con la interfaz.	1 día		26/04/2021	26/04/2021
<b>Historia N°10: Mantenimiento de las entradas de productos</b>	<b>RF21, RF22</b>	4 días	27/04/2021	30/04/2021
Elaboración de procedimientos para el CRUD de datos.		1 día	27/04/2021	27/04/2021
Integración del código a la interfaz del módulo.		2 días	28/04/2021	29/04/2021
Comprobación de los campos de texto para el CRUD.		1 día	30/04/2021	30/04/2021

En la tabla anterior se muestra el cronograma del Sprint 3, en donde se muestra las fechas en las que se realizarán los requerimientos funcionales con respecto a cada una de sus historias de usuario.

## Historia N°8: Mantenimiento de los productos

Elaboración de procedimientos para el CRUD de datos.

Figura N°152. Producto - Registrar

```
57 public function store(Request $request)
58 {
59     $request->validate([
60         'nombre' => ['required', 'string', 'max:255'],
61         'marca_id' => ['required'],
62         'precio' => ['required', 'regex:/^[0-9]+(\.[0-9]{0-2})?$/ ', 'max:10'],
63         'cantidad' => ['required', 'numeric', 'min:1', 'max:600'],
64         'cantidad_max' => ['required', 'numeric', 'min:1', 'max:600'],
65     ]);
66
67     $productos = new ProductoModel([
68         'nombre' => $request->get('nombre'),
69         'marca_id' => $request->get('marca_id'),
70         'precio' => $request->get('precio')
71     ]);
72
73     $productos->save();
74
75     $stock_productos = new StockProductoModel([
76         'producto_id' => $productos->id,
77         'cantidad' => $request->get('cantidad'),
78         'cantidad_max' => $request->get('cantidad_max')
79     ]);
80
81     $stock_productos->save();
82
83     return redirect('/producto')->with('success', 'Producto guardado');
84 }
```

Figura N°153. Producto - Editar

```
118 public function update(Request $request, $id)
119 {
120     $request->validate([
121         'nombre' => ['required', 'string', 'max:255'],
122         'marca_id' => ['required'],
123         'precio' => ['required', 'regex:/^[0-9]+(\.[0-9]{0-2})?$/ ', 'max:10'],
124         'cantidad' => ['required', 'numeric', 'min:1', 'max:600'],
125         'cantidad_max' => ['required', 'numeric', 'min:1', 'max:600'],
126     ]);
127
128     $producto_update = ProductoModel::find($id);
129     $producto_update->nombre = $request->get('nombre');
130     $producto_update->marca_id = $request->get('marca_id');
131     $producto_update->precio = $request->get('precio');
132     $producto_update->save();
133
134     $stock_productos_update = StockProductoModel::where('producto_id',$id)->first();
135     $stock_productos_update->cantidad = $request->get('cantidad');
136     $stock_productos_update->cantidad_max = $request->get('cantidad_max');
137     $stock_productos_update->save();
138
139     return redirect('/producto')->with('success', 'Producto actualizado!');
140 }
```

Figura N°154. Producto - Eliminar

```
148 public function destroy($id)
149 {
150     $producto_eliminar = ProductoModel::find($id);
151     $producto_eliminar->delete();
152
153     $stock_productos_eliminar = StockProductoModel::where('producto_id',$id)->first();
154     $stock_productos_eliminar->delete();
155
156     return redirect('/producto')->with('success', 'Producto eliminado!');
157 }
158 }
```

Integración del código a la interfaz del módulo.

Figura N°155. Producto - Interfaz

```
1 @extends('inicio')
2
3 @section('main')
4 <div class="row">
5 <h2>Productos</h2>
6 <div class="col-sm-12">
7 <table id="tabla_general" class="table table-striped">
8 <thead>
9 <tr>
10 <th>N°</th>
11 <th>Nombre</th>
12 <th>Marca</th>
13 <th>Precio</th>
14 <th>Stock</th>
15 <th>Stock Límite</th>
16 <th>Stock Exceso</th>
17 <th>Acciones</th>
18 </tr>
19 </thead>
20 <tbody>
21 <?php $i=1 ?>
22 <?php $b=1 ?>
23 @foreach($productos as $prod)
24 <tr>
25 <td><span>{{ $i++ }}</span></td>
26 <td><span>{{ $prod->nombre }}</span></td>
27 <td><span>{{ $prod->marca }}</span></td>
28 <td><span>{{ number_format($prod->precio,2) }}</span></td>
```

Comprobación de los campos de texto para el CRUD.

Figura N°156. Producto – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4 <div class="container">
5 <div class="row justify-content-center">
6 <div class="col-md-8">
7 <div class="card">
8 <div class="card-header">{{ __('Register') }}</div>
9 <div class="card-body">
10 <form method="POST" action="{{ route('producto.store') }}">
11 @csrf
12 <div class="form-group row">
13 <label for="nombre" class="col-md-4 col-form-label text-md-right">{{ __('Nombre') }}</label>
14 <div class="col-md-6">
15 <input id="nombre" type="text" class="form-control @error('nombre') is-invalid @enderror" name="nombre" value="{{ old('nombre') }}" required autocomplete="nombre" autofocus>
16 @error('nombre')
17 <span class="invalid-feedback" role="alert">
18 <strong>{{ $message }}</strong>
19 </span>
20 @enderror
21 </div>
22 </div>
23 <div class="form-group row">
24 <label for="marca" class="col-md-4 col-form-label text-md-right">{{ __('Marca') }}</label>
25 <div class="col-md-6">
26 <select class="form-control @error('marca_id') is-invalid @enderror" name="marca_id" required>
27 <option value="{{ null }}">-- Seleccionar --</option>
28 @foreach ($marcas as $marca)
29 <option value="{{ $marca->id }}">{{ $marca->nombre }}</option>
30 @endforeach
31 </select>
32 @error('marca')
33 <span class="invalid-feedback" role="alert">
34 <strong>{{ $message }}</strong>
35 </span>
36 @enderror
37 </div>
38 </div>
39 <div class="form-group row">
40 <label for="precio" class="col-md-4 col-form-label text-md-right">{{ __('Precio') }}</label>
41 <div class="col-md-6">
42 <input id="precio" type="text" class="form-control @error('precio') is-invalid @enderror" name="precio" value="{{ old('precio') }}" required autocomplete="precio" autofocus>
43 </div>
44 </div>
```

## Historia N°9: Alerta de stock mínimo

Elaboración del procedimiento para obtener los datos.

Figura N°157. Alerta de stock - Script

```

29     @php
30     $rojo    = ($prod->cantidad_max*25)/100;
31     $naranja = ($prod->cantidad_max*50)/100;
32     @endphp
33     <td class="cantidad_p">
34     @if($prod->cantidad > $naranja)
35     <span class="verde">{{ $prod->cantidad }}</span>
36     @elseif($prod->cantidad <= $naranja && $prod->cantidad > $rojo)
37     <span class="naranja">{{ $prod->cantidad }}</span>
38     @elseif($prod->cantidad <= $rojo)
39     <span class="rojo">{{ $prod->cantidad }}</span>
40     @endif
41     </td>
42     <td class="cantidad_p"><span>{{ $prod->cantidad_max }}</span></td>
43     <td class="cantidad_p">
44     @if($prod->cantidad > $prod->cantidad_max)
45     <span class="rojo">{{ $prod->cantidad - $prod->cantidad_max }}</span>
46     @else
47     <span>0</span>
48     @endif
49     </td>

```

Integración del código con la interfaz.

Figura N°158. Alerta de stock – Diseño

```

150 body#sistema_body .sistema_contenido .contenido_general .vista table td.cantidad_p span{
151     left: 50%;
152     transform: translateX(-50%);
153 }
154 body#sistema_body .sistema_contenido .contenido_general .vista table td span{
155     display: flex;
156     align-items: center;
157     position: absolute;
158     height: 100%;
159 }
160 body#sistema_body .sistema_contenido .contenido_general .vista table td span.rojo{
161     background-color: #f70a0a;
162     width: 30%;
163     color: #fff;
164     border-radius: 5px;
165     justify-content: center;
166 }
167 body#sistema_body .sistema_contenido .contenido_general .vista table td span.naranja{
168     background-color: #f3a20f;
169     width: 30%;
170     color: #fff;
171     border-radius: 5px;
172     justify-content: center;
173 }
174 body#sistema_body .sistema_contenido .contenido_general .vista table td span.verde{
175     background-color: #3c944d;
176     width: 30%;
177     color: #fff;
178     border-radius: 5px;
179     justify-content: center;
180 }

```

Figura N°159. Alerta de stock – Muestra de datos

N°	Nombre	Marca	Precio	Stock	Stock Límite	Stock Exceso	Acciones
1	Balón de 5kg	Zeta Gas	18.00	3	20	0	 
2	Balón de 10kg	Zeta Gas	40.00	6	40	0	 
3	Balón de 45kg	Zeta Gas	168.00	6	30	0	 
4	Balón de 10kg vacío	Zeta Gas	110.00	7	20	0	 

## Historia N°10: Mantenimiento de las entradas de productos

Elaboración de procedimientos para el CRUD de datos.

Figura N°160. Entrada de producto - Registrar

```
62 public function store(Request $request)
63 {
64     $request->validate([
65         'proveedor_id'     => ['required', 'string', 'max:255'],
66         'producto_id'     => ['required'],
67         'cantidad'        => ['required', 'numeric', 'min:1', 'max:600'],
68         'monto_unidad'    => ['required', 'regex:/^[0-9]+(\.[0-9]{0-9})?$/ ', 'max:10'],
69     ]);
70
71     $monto_total = $request->get('cantidad') * $request->get('monto_unidad');
72     $user = auth()->user()->id;
73
74     $pedidos = new PedidoModel([
75         'user_id'         => $user,
76         'proveedor_id'    => $request->get('proveedor_id'),
77     ]);
78
79     $pedidos->save();
80
81     $detalle_pedidos = new DetallePedidoModel([
82         'pedido_id'       => $pedidos->id,
83         'producto_id'     => $request->get('producto_id'),
84         'cantidad'        => $request->get('cantidad'),
85         'monto_unidad'    => $request->get('monto_unidad'),
86         'monto_total'     => $monto_total
87     ]);
88
89     $detalle_pedidos->save();
90
91     $stock_producto = StockProductoModel::select('cantidad')->where('producto_id', $detalle_pedidos->producto_id)->first();
92     $nuevo_stock = $stock_producto->cantidad + $detalle_pedidos->cantidad;
93
94     $actualizar_stock = StockProductoModel::where('producto_id', $detalle_pedidos->producto_id)
95         ->update(['cantidad' => $nuevo_stock]);
96
97     return redirect('/pedido')->with('success', 'Entrada registrada');
98 }
```

Figura N°161. Entrada de producto - Editar

```
133 public function update(Request $request, $id)
134 {
135     $request->validate([
136         'proveedor_id'     => ['required', 'string', 'max:255'],
137         'producto_id'     => ['required'],
138         'cantidad'        => ['required', 'numeric', 'min:1', 'max:600'],
139         'monto_unidad'    => ['required', 'regex:/^[0-9]+(\.[0-9]{0-9})?$/ ', 'max:10'],
140     ]);
141
142     $pedido_actual = DetallePedidoModel::select('cantidad', 'producto_id')->where('pedido_id', $id)->first();
143     $stock_producto_actual = StockProductoModel::select('cantidad')
144         ->where('producto_id', $pedido_actual->producto_id)->first();
145     $stock_producto_antiguo = $stock_producto_actual->cantidad - $pedido_actual->cantidad;
146     $monto_total = $request->get('cantidad') * $request->get('monto_unidad');
147
148     $pedido_update = PedidoModel::find($id);
149     $pedido_update->proveedor_id = $request->get('proveedor_id');
150     $pedido_update->save();
151
152     $detalle_pedido_update = DetallePedidoModel::where('pedido_id', $id)->first();
153     $detalle_pedido_update->producto_id = $request->get('producto_id');
154     $detalle_pedido_update->cantidad = $request->get('cantidad');
155     $detalle_pedido_update->monto_unidad = $request->get('monto_unidad');
156     $detalle_pedido_update->monto_total = $monto_total;
157     $detalle_pedido_update->save();
158
159     if($pedido_actual->producto_id == $detalle_pedido_update->producto_id){
160
161         $stock_producto_nuevo = $stock_producto_antiguo + $detalle_pedido_update->cantidad;
162         $actualizar_stock = StockProductoModel::where('producto_id', $detalle_pedido_update->producto_id)
163             ->update(['cantidad' => $stock_producto_nuevo]);
164     }else{
165         $antiguo_stock = StockProductoModel::where('producto_id', $pedido_actual->producto_id)
166             ->update(['cantidad' => $stock_producto_antiguo]);
167         $stock_producto = StockProductoModel::select('cantidad')
168             ->where('producto_id', $detalle_pedido_update->producto_id)->first();
169         $nuevo_stock = $stock_producto->cantidad + $detalle_pedido_update->cantidad;
170         $actualizar_stock = StockProductoModel::where('producto_id', $detalle_pedido_update->producto_id)
171             ->update(['cantidad' => $nuevo_stock]);
172     }
173
174     return redirect('/pedido')->with('success', 'Entrada actualizada!');
175 }
176 }
```



Figura N°162. Entrada de producto - Eliminar

```

184 public function destroy($id)
185 {
186     $pedido_actual = DetallePedidoModel::select('cantidad', 'producto_id')->where('pedido_id',$id)->first();
187     $stock_producto_actual = StockProductoModel::select('cantidad')
188     ->where('producto_id', $pedido_actual->producto_id)->first();
189     $stock_producto_antiguo = $stock_producto_actual->cantidad - $pedido_actual->cantidad;
190
191     $pedido_eliminar = PedidoModel::find($id);
192     $pedido_eliminar->delete();
193
194     $detalle_pedidos_eliminar = DetallePedidoModel::where('pedido_id',$id);
195     $detalle_pedidos_eliminar->delete();
196
197     $antiguo_stock = StockProductoModel::where('producto_id', $pedido_actual->producto_id)
198     ->update(['cantidad' => $stock_producto_antiguo]);
199
200     return redirect('/pedido')->with('success', 'Entrada eliminada!');
201 }

```

Integración del código a la interfaz del módulo.

Figura N°163. Entrada de producto - Interfaz

```

1 @extends('inicio')
2
3 @section('main')
4 <div class="row">
5 <h2>Entrada de productos</h2>
6 <div class="col-sm-12">
7 <table id="tabla_general" class="table table-striped">
8 <thead>
9 <tr>
10 <th>ID</th>
11 <th>Proveedor</th>
12 <th>Producto</th>
13 <th>Cantidad</th>
14 <th>Precio Unidad</th>
15 <th>Precio Total</th>
16 <th>Fecha</th>
17 <th>Acciones</th>
18 </tr>
19 </thead>
20 <tbody>
21 <tr>
22 <td>
23 @foreach($pedidos as $pe)
24 <tr>
25 <td><span>{{ $i++ }}</span></td>
26 <td><span>{{ $pe->proveedor }}</span></td>
27 <td><span>{{ $pe->producto }}</span></td>
28 <td><span>{{ $pe->cantidad }}</span></td>
29 <td><span>{{ number_format($pe->monto_unidad,2) }}</span></td>
30 <td><span>{{ number_format($pe->monto_total,2) }}</span></td>
31 <td><span>{{ \Carbon\Carbon::parse($pe->created_at)->formatLocalized('%Y-%m-%d') }}</span></td>
32 <td>
33 <div class="botones">
34 <a class="accion_editar" data-bs-toggle="tooltip" title="Editar" href="{{ route('pedido.edit',$pe->id) }}" >
35 <form id="{{ $pe->id }}" action="{{ route('pedido.destroy',$pe->id) }}" method="post">
36 @csrf
37 @method('DELETE')
38 <button type="button" id="{{ $pe->id }}" class="accion_eliminar" onclick="eliminarRegistro(event)">
39 </form>
40 </div>

```

Comprobación de los campos de texto para el CRUD.

Figura N°164. Entrada de producto – Ingreso de datos

```

1 @extends('inicio')
2
3 @section('main')
4 <div class="container">
5 <div class="row justify-content-center">
6 <div class="col-md-8">
7 <div class="card">
8 <div class="card-header">{{ __('Register') }}</div>
9
10 <div class="card-body">
11 <form method="POST" action="{{ route('pedido.store') }}">
12 @csrf
13
14 <div class="form-group row">
15 <label for="proveedor_id" class="col-md-4 col-form-label text-md-right">{{ __('Proveedor') }}</label>
16
17 <div class="col-md-6">
18 <select class="form-control @error('proveedor_id') is-invalid @enderror" name="proveedor_id" required>
19 <option value="{{ null }}">-- Seleccionar --</option>
20 @foreach($proveedores as $prov)
21 <option value="{{ $prov->id }}">{{ $prov->nombre }}</option>
22 @endforeach
23 </select>
24
25 @error('proveedor_id')
26 <span class="invalid-feedback" role="alert">
27 <strong>{{ $message }}</strong>
28 </span>
29 @enderror
30 </div>
31 </div>
32
33 <div class="form-group row">
34 <label for="producto_id" class="col-md-4 col-form-label text-md-right">{{ __('Producto') }}</label>
35
36 <div class="col-md-6">
37 <select class="form-control @error('producto_id') is-invalid @enderror" name="producto_id" required>
38 <option value="{{ null }}">-- Seleccionar --</option>
39 @foreach($productos as $pro)
40 <option value="{{ $pro->id }}">{{ $pro->nombre }}</option>
41 @endforeach
42 </select>
43
44 @error('producto_id')
45 <span class="invalid-feedback" role="alert">
46 <strong>{{ $message }}</strong>
47 </span>
48 @enderror
49 </div>
50 </div>
51 </div>

```

### Acta de entrega del Sprint 3

Siendo las 18:00 horas del día 30 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Lesly Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura a los requerimientos realizados y encargados del Sprint 3, asimismo muestra las interfaces y el código realizado hasta el momento.

Verificadas las explicaciones y el sustento presentado por el señor Romero Rodríguez, Billy Jerson para la aprobación del Sprint 3, se decide aprobar por parte de la gerente general Estupiñan Palma, Patricia el término del Sprint 3, del proyecto "Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C".



Firma

## Sprint 4

A continuación, se mostrarán los requerimientos funcionales que se realizarán para el sprint 4.

Tabla N°39: *Sprint 4*

#	Requerimientos Funcionales	N° Historia	Nombre Historia	Prioridad	Tiempo
1	Permita registrar, editar y eliminar registros de las salidas de los productos.	H11	Mantenimiento de las salidas de productos	Media	5
	Permita exportar los datos.				
2	Permita registrar, editar y eliminar registros de la lista de una salida de productos.	H12	Lista de productos de una salida de productos	Media	3
	Permita agregar otros productos.				
	Permita exportar los datos.				
3	Permita registrar, editar y eliminar registros de préstamos.	H13	Mantenimiento de los préstamos	Media	3
	Muestre si se realizó la devolución.				
	Permita exportar los datos.				
Total					11

#### Planificación del Sprint 4

Siendo las 8:30 horas del día 01 de mayo del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura de los requerimientos del sistema.

Analizada la lista de requerimientos por el señor Romero Rodríguez Billy Jerson solventa algunas dudas sobre los requerimientos y se compromete a cumplir con los requisitos establecidos por el Sprint 4.

Los presentes imparten su aprobación a los requerimientos de acuerdo a lo establecido en la planificación del Sprint 4 indicando que la fecha de entrega de este Sprint sería el día 11 de mayo de 2021.



Firma

## Cronograma de tareas del Sprint 4

Tabla N°40: Cronograma del Sprint 4

<b>Sprint 4</b>	<b>Requerimiento F.</b>	01/05/2021	11/05/2021	01/05/2021
<b>Historia N°11: Mantenimiento de las salidas de productos</b>	<b>RF23, RF24</b>	01/05/2021	05/05/2021	01/05/2021
Elaboración de procedimientos para el CRUD de datos.		01/05/2021	02/05/2021	01/05/2021
Integración del código a la interfaz del módulo.		03/05/2021	04/05/2021	03/05/2021
Comprobación de los campos de texto para el CRUD.		05/05/2021	05/05/2021	05/05/2021
<b>Historia N°12: Lista de productos de una salida de productos</b>		<b>RF25, RF26, RF27</b>	06/05/2021	08/05/2021
Elaboración de procedimientos para el CRUD de datos.	06/05/2021		06/05/2021	06/05/2021
Integración del código a la interfaz del módulo.	07/05/2021		07/05/2021	07/05/2021
Comprobación de los campos de texto para el CRUD.	08/05/2021		08/05/2021	08/05/2021
<b>Historia N°13: Mantenimiento de los préstamos</b>	<b>RF28, RF29, RF30</b>	09/05/2021	11/05/2021	09/05/2021
Elaboración de procedimientos para el CRUD de datos.		09/05/2021	09/05/2021	09/05/2021
Integración del código a la interfaz del módulo.		10/05/2021	10/05/2021	10/05/2021
Comprobación de los campos de texto para el CRUD.		11/05/2021	11/05/2021	11/05/2021

En la tabla anterior se muestra el cronograma del Sprint 4, en donde se muestra las fechas en las que se realizarán los requerimientos funcionales con respecto a cada una de sus historias de usuario.

## Historia N°11: Mantenimiento de las salidas de productos

Elaboración de procedimientos para el CRUD de datos.

Figura N°165. Salida de producto - Registrar

```
56 public function store(Request $request)
57 {
58     $request->validate([
59         'cliente_id' => ['required'],
60     ]);
61
62     $user= auth()->user()->id;
63
64     $salidas = new SalidaModel([
65         'user_id' => $user,
66         'cliente_id' => $request->get('cliente_id'),
67     ]);
68
69     $salidas->save();
70
71     $id = $salidas->id;
72
73     $detalle_salidas = DB::table('detalle_salidas as ds')
74         ->select('ds.id','ds.salida_id','pro.id as producto_id','pro.nombre','pro.precio','ds.cantidad')
75         ->join('productos as pro','ds.producto_id','=','pro.id')
76         ->join('stock_productos as sp','sp.producto_id','=','pro.id')
77         ->where('ds.salida_id',$id)
78         ->get();
79
80
81     return view('detalle_salida.index', compact('detalle_salidas','id'));
82 }
```

Figura N°166. Salida de producto - Editar

```
116 public function update(Request $request, $id)
117 {
118     $request->validate([
119         'cliente_id' => ['required'],
120     ]);
121
122     $salida_update = SalidaModel::find($id);
123     $salida_update->cliente_id = $request->get('cliente_id');
124     $salida_update->save();
125
126     return redirect('/salida')->with('success', 'Salida actualizada!');
127 }
```

Figura N°167. Salida de producto - Eliminar

```
135 public function destroy($id)
136 {
137     $detalle_salidas = DetalleSalidaModel::select('producto_id','cantidad')->where('salida_id',$id)->get();
138
139     for ($i=0; $i<count($detalle_salidas); $i++) {
140         $stock_producto = StockProductoModel::select('id','cantidad')
141             ->where('producto_id',$detalle_salidas[$i]->producto_id)->first();
142         $cantidad_restaurada = $detalle_salidas[$i]->cantidad + $stock_producto->cantidad;
143         $actualizar_stock = StockProductoModel::where('id',$stock_producto->id)
144             ->update(['cantidad' => $cantidad_restaurada]);
145     }
146
147     $salida_eliminar = SalidaModel::find($id);
148     $salida_eliminar->delete();
149
150     $detalle_salida_eliminar = DetalleSalidaModel::where('salida_id',$id)->delete();
151
152
153     return redirect('/salida')->with('success', 'Salida eliminada!');
154 }
```

## Integración del código a la interfaz del módulo.

Figura N°168. Salida de producto - Interfaz

```
1 @extends('inicio')
2
3 @section('main')
4 <div class="loading"></div>
5 <div class="row">
6 <h2>Salida de productos</h2>
7 <div class="col-sm-12">
8 <table id="tabla_general" class="table table-striped">
9 <thead>
10 <tr>
11 <th>N°</th>
12 <th>Clientes</th>
13 <th>Precio Total</th>
14 <th>Fecha</th>
15 <th>Acciones</th>
16 </tr>
17 </thead>
18 <tbody>
19 <?php $i=1 ?>
20 @foreach($salidas as $sa)
21 <tr>
22 <td><span>{{ $i+ }}</span></td>
23 <td><span>{{ $sa->nombre }} {{ $sa->apellido_paterno }} {{ $sa->apellido_materno }}</span></td>
24 <td><span>{{ number_format($sa->monto_total,2) }}</span></td>
25 <td><span>{{ \Carbon\Carbon::parse($sa->created_at)->formatLocalized('%Y-%m-%d') }}</span></td>
26 <td>
27 <div class="botones">
28 <a class="accion_editar" data-bs-toggle="tooltip" title="Editar" href="{{ route('salida.edit',$sa->id) }}" >i class="fas fa-pen-square"</i></a>
29 <a class="accion_editar" data-bs-toggle="tooltip" title="Editar" href="{{ route('detalle_salida.show',$sa->id) }}">i class="fas fa-list-alt"</i></a>
30 <form id="{{ $sa->id }}" action="{{ route('salida.destroy', $sa->id) }}" method="post">
31 @csrf
32 @method('DELETE')
33 <button type="button" id="{{ $sa->id }}" class="accion_eliminar" onclick="eliminarRegistro(event)">i class="fas fa-trash-alt"</i></button>
34 </form>
35 </div>
36 </td>
37 </tr>
38 @endforeach
39 </tbody>
40 </table>
41 </div>
42 <a style="margin: 5px 0px 0px 0px;" href="{{ route('salida.create') }}" class="btn btn-primary">Nueva salida</a>
43
44 @if (session('success'))
45 <div class="alert alert-success">
46 {{ session('success') }}
47 </div>
48 @endif
49
```

## Comprobación de los campos de texto para el CRUD.

Figura N°169. Salida de producto – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4
5 <div class="container">
6 <div class="row justify-content-center">
7 <div class="col-md-8">
8 <div class="card">
9 <div class="card-header">{{ __('Register') }}</div>
10
11 <div class="card-body">
12 <form method="POST" action="{{ route('salida.store') }}">
13 @csrf
14
15 <div class="form-group row">
16 <label for="cliente_id" class="col-md-4 col-form-label text-md-right">{{ __('Cliente') }}</label>
17
18 <div class="col-md-6">
19 <select class="form-control @error('cliente_id') is-invalid @enderror" name="cliente_id" required>
20 <option value="{{ null }}">-- Seleccionar --</option>
21 @foreach ($clientes as $cliente)
22 <option value="{{ $cliente->id }}">{{ $cliente->nombre }} {{ $cliente->apellido_paterno }} {{ $cliente->
23 apellido_materno }}</option>
24 @endforeach
25 </select>
26
27 @error('cliente_id')
28 <span class="invalid-feedback" role="alert">
29 <strong>{{ $message }}</strong>
30 </span>
31 @enderror
32 </div>
33
34 <div class="form-group row mb-0">
35 <div class="col-md-6 offset-md-4">
36 <button type="submit" class="btn btn-primary">
37 {{ __('Apregar productos') }}
38 </button>
39 </div>
40 </div>
41 </form>
42 </div>
43 </div>
44 </div>
45 </div>
46
```

## Historia N°12: Lista de productos de una salida de productos

Elaboración de procedimientos para el CRUD de datos.

Figura N°170. Lista de productos – Registrar

```
54 public function store(Request $request)
55 {
56     $request->validate([
57         'salida_id' => ['required'],
58         'producto_id' => ['required'],
59         'cantidad' => ['required', 'numeric', 'min:1', 'max:600'],
60     ]);
61
62     $d_salidas = new DetallesSalidaModel([
63         'salida_id' => $request->get('salida_id'),
64         'producto_id' => $request->get('producto_id'),
65         'cantidad' => $request->get('cantidad'),
66     ]);
67
68     $d_salidas->save();
69
70     $cantidad_total = DB::table('detalle_salidas as ds')
71         ->select(DB::raw('ds.cantidad * pro.precio as monto_unidad'))
72         ->join('productos as pro', 'ds.producto_id', '=', 'pro.id')
73         ->where('ds.salida_id', $d_salidas->salida_id)
74         ->whereNull('ds.deleted_at')
75         ->get();
76
77     $suma=0;
78
79     for ($i=0; $i <count($cantidad_total) ; $i++) {
80         $suma = $cantidad_total[$i]->monto_unidad + $suma;
81     }
82
83     $actualizar_monto_total = SalidaModel::where('id', $d_salidas->salida_id)->update(['monto_total' => $suma]);
84
85     $actual_stock = StockProductoModel::select('cantidad')->where('producto_id', $d_salidas->producto_id)->first();
86     $resta_stock = $actual_stock->cantidad - $d_salidas->cantidad;
87     $actualizar_stock = StockProductoModel::where('producto_id', $d_salidas->producto_id)
88         ->update(['cantidad' => $resta_stock]);
89
90     $id = $d_salidas->salida_id;
91     $detalle_salidas = DB::table('detalle_salidas as ds')
92         ->select('ds.id', 'ds.salida_id', 'pro.id as producto_id', 'pro.nombre', 'pro.precio', 'ds.cantidad')
93         ->join('productos as pro', 'ds.producto_id', '=', 'pro.id')
94         ->join('stock_productos as sp', 'sp.producto_id', '=', 'pro.id')
95         ->where('ds.salida_id', $d_salidas->salida_id)
96         ->whereNull('ds.deleted_at')
97         ->get();
98
99     return view('detalle_salida.index', compact('detalle_salidas', 'id'));
100 }
```

Figura N°171. Lista de productos – Editar

```
150 public function update(Request $request, $id)
151 {
152     $request->validate([
153         'producto_id' => ['required'],
154         'cantidad' => ['required', 'numeric', 'min:1', 'max:600'],
155     ]);
156
157     $d_salida_actual = DetallesSalidaModel::select('cantidad', 'producto_id')->where('id', $id)->first();
158
159     if($d_salida_actual->producto_id == $request->get('producto_id')){
160         $stock_producto_actual = StockProductoModel::select('cantidad')
161             ->where('producto_id', $d_salida_actual->producto_id)->first();
162         $stock_producto_antiguo = $stock_producto_actual->cantidad + $d_salida_actual->cantidad;
163
164         $d_salidas_update = DetallesSalidaModel::find($id);
165         $d_salidas_update->producto_id = $request->get('producto_id');
166         $d_salidas_update->cantidad = $request->get('cantidad');
167         $d_salidas_update->save();
168
169         $cantidad_total = DB::table('detalle_salidas as ds')
170             ->select(DB::raw('ds.cantidad * pro.precio as monto_unidad'))
171             ->join('productos as pro', 'ds.producto_id', '=', 'pro.id')
172             ->where('ds.salida_id', $d_salidas_update->salida_id)
173             ->whereNull('ds.deleted_at')
174             ->get();
175
176         $suma=0;
177
178         for ($i=0; $i <count($cantidad_total) ; $i++) {
179             $suma = $cantidad_total[$i]->monto_unidad + $suma;
180         }
181
182         $actualizar_monto_total = SalidaModel::where('id', $d_salidas_update->salida_id)->update(['monto_total' => $suma]);
183
184         $stock_producto_nuevo = $stock_producto_antiguo - $d_salidas_update->cantidad;
185         $actualizar_stock = StockProductoModel::where('producto_id', $d_salidas_update->producto_id)
186             ->update(['cantidad' => $stock_producto_nuevo]);
187
188         $detalle_salidas = DB::table('detalle_salidas as ds')
189             ->select('ds.id', 'ds.salida_id', 'pro.id as producto_id', 'pro.nombre', 'pro.precio', 'ds.cantidad')
190             ->join('productos as pro', 'ds.producto_id', '=', 'pro.id')
191             ->join('stock_productos as sp', 'sp.producto_id', '=', 'pro.id')
192             ->where('ds.salida_id', $d_salidas_update->salida_id)
193             ->whereNull('ds.deleted_at')
194             ->get();
195
196         $id = $d_salidas_update->salida_id;
197
198         return view('detalle_salida.index', compact('detalle_salidas', 'id'));
199     }else{
200         $stock_producto_actual = StockProductoModel::select('cantidad')
201             ->where('producto_id', $d_salida_actual->producto_id)->first();
202         $stock_producto_antiguo = $stock_producto_actual->cantidad + $d_salida_actual->cantidad;
```



Figura N°172. Lista de productos – Eliminar

```

254 public function destroy($id)
255 {
256     $d_salida_actual = DetalleSalidaModel::select('cantidad','producto_id','salida_id')->where('id',$id)->first();
257     $stock_producto_actual = StockProductoModel::select('cantidad')
258     ->where('producto_id',$d_salida_actual->producto_id)->first();
259     $stock_producto_antiguo = $stock_producto_actual->cantidad + $d_salida_actual->cantidad;
260
261     $d_salidas_eliminar = DetalleSalidaModel::find($id);
262     $d_salidas_eliminar->delete();
263
264     $cantidad_total = DB::table('detalle_salidas as ds')
265     ->select(DB::raw("ds.cantidad * pro.precio as monto_unidad"))
266     ->join('productos as pro','ds.producto_id','=','pro.id')
267     ->where('ds.salida_id',$d_salida_actual->salida_id)
268     ->whereNull('ds.deleted_at')
269     ->get();
270
271     $suma=0;
272
273     for ($i=0; $i <count($cantidad_total) ; $i++) {
274         $suma = $cantidad_total[$i]->monto_unidad + $suma;
275     }
276
277     $actualizar_monto_total = SalidaModel::where('id',$d_salida_actual->salida_id)->update(['monto_total' => $suma]);
278     $actualizar_stock = StockProductoModel::where('producto_id',$d_salida_actual->producto_id)
279     ->update(['cantidad' => $stock_producto_antiguo]);
280
281     $detalle_salidas = DB::table('detalle_salidas as ds')
282     ->select('ds.id','ds.salida_id','pro.id as producto_id','pro.nombre','pro.precio','ds.cantidad')
283     ->join('productos as pro','ds.producto_id','=','pro.id')
284     ->join('stock_productos as sp','sp.producto_id','=','pro.id')
285     ->where('ds.salida_id',$d_salida_actual->salida_id)
286     ->whereNull('ds.deleted_at')
287     ->get();
288
289     $id = $d_salida_actual->salida_id;
290
291     return view('detalle_salida.index', compact('detalle_salidas','id'));
292 }

```

Integración del código a la interfaz del módulo.

Figura N°173. Lista de productos – Interfaz

```

1 @extends('inicio')
2
3 @section('main')
4 <div class="row">
5 <h2>Lista de productos</h2>
6 <div class="col-sm-12">
7 <table id="tabla_general" class="table table-striped">
8 <thead>
9 <tr>
10 <th>Nº</th>
11 <th>Producto</th>
12 <th>Precio</th>
13 <th>Cantidad</th>
14 <th>Acciones</th>
15 </tr>
16 </thead>
17 <tbody>
18 <tr>
19 @foreach($detalle_salidas as $ds)
20 <tr>
21 <td><span>{{ $i++ }}</span></td>
22 <td><span>{{ $ds->nombre }}</span></td>
23 <td><span>{{ number_format($ds->precio,2) }}</span></td>
24 <td><span>{{ $ds->cantidad }}</span></td>
25 <td>
26 <div class="botones">
27 <a class="accion_editar" data-bs-toggle="tooltip" title="Editar" href="{{ route('detalle_salida.edit',$ds->id) }}" >i class="fas fa-pen-square"
28 </i></a>
29 <form id="{{ $ds->id }}" action="{{ route('detalle_salida.destroy', $ds->id) }}" method="post">
30 @csrf
31 <button type="button" id="{{ $ds->id }}" class="accion_eliminar" onclick="eliminarRegistro(event)">i class="fas fa-trash-alt">/i</button>
32 </form>
33 </div>
34 </td>
35 </tr>
36 @endforeach
37 </tbody>
38 </table>
39 </div>
40 </div>
41 <a style="margin: 5px 0px 0px 0px;" href="{{ route('salida.index') }}" class="btn btn-primary">Atras</a>
42 <a style="margin: 5px 0px 0px 15px;" href="{{ route('agregar',$id) }}" class="btn btn-primary">Agregar producto</a>

```

Comprobación de los campos de texto para el CRUD.

Figura N°174. Lista de productos – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4
5 <div class="container">
6 <div class="row justify-content-center">
7 <div class="col-md-8">
8 <div class="card">
9 <div class="card-header">
10 <h3>{{ __('Register') }}</h3>
11 </div>
12 <div class="card-body">
13 <form id="verificarsp" method="POST" action="{{ route('detalle_salida.store') }}">
14 @csrf
15
16 <input class="d-none" type="text" name="salida_id" value="{{ $salida_id }}" />
17
18 <div class="form-group row">
19 <label for="producto_id" class="col-md-4 col-form-label text-md-right">{{ __('Producto') }}</label>
20
21 <div class="col-md-6">
22 <select id="producto_id" class="form-control @error('producto_id') is-invalid @enderror" name="producto_id" required>
23 <option value="{ null }">-- Seleccionar --</option>
24 @foreach ($productos as $pro)
25 <option value="{{ $pro->id }}">{{ $pro->nombre }}</option>
26 @endforeach
27 </select>
28
29 @error('producto_id')
30 <span class="invalid-feedback" role="alert">
31 <strong>{{ $message }}</strong>
32 </span>
33 @enderror
34 </div>
35 </div>
36
37 <div class="form-group row monitoreo">
38 <label class="col-md-4 col-form-label text-md-right">Stock producto:</label>
39 <div class="col-md-6">
40 <input id="stock" class="form-control" type="text" value="" disabled>
41 </div>
42 </div>
43
44 <div class="form-group row">
45 <label for="cantidad" class="col-md-4 col-form-label text-md-right">{{ __('Cantidad') }}</label>
46
47 <div class="col-md-6">
48 <input id="cantidad" type="text" class="form-control @error('cantidad') is-invalid @enderror" name="cantidad" value="{{ old('cantidad') }}" required autocomplete="cantidad" autofocus>
49
50 @error('cantidad')
51 <span class="invalid-feedback" role="alert">
52 <strong>{{ $message }}</strong>
53 </span>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
```

## Historia N°13: Mantenimiento de los préstamos

Elaboración de procedimientos para el CRUD de datos.

Figura N°175. Prestamos - Registrar

```
61 public function store(Request $request)
62 {
63     $request->validate([
64         'cliente_id' => ['required'],
65         'producto_id' => ['required'],
66         'cantidad' => ['required', 'numeric', 'min:1', 'max:600'],
67     ]);
68
69     $user= auth()->user()->id;
70
71     $prestamos = new PrestamoModel([
72         'user_id' => $user,
73         'cliente_id' => $request->get('cliente_id'),
74         'producto_id' => $request->get('producto_id'),
75         'cantidad' => $request->get('cantidad')
76     ]);
77
78     $prestamos->save();
79
80     $stock_producto = StockProductoModel::select('producto_id','cantidad')
81         ->where('producto_id',$prestamos->producto_id)->first();
82     $stock_producto_nuevo = $stock_producto->cantidad - $prestamos->cantidad;
83
84     $actualizar_stock = StockProductoModel::where('producto_id',$prestamos->producto_id)
85         ->update(['cantidad' => $stock_producto_nuevo]);
86
87     return redirect('/prestamo')->with('success', 'Prestamo resgistrado');
88 }
```

Figura N°176. Prestamos - Editar

```

134 public function update(Request $request, $id)
135 {
136     $cantidad_prestamo = PrestamoModel::select('producto_id','cantidad')->where('id',$id)->first();
137
138     $request->validate([
139         'cliente_id' => ['required'],
140         'producto_id' => ['required'],
141         'cantidad' => ['required', 'numeric', 'min:1', 'max:600'],
142         'retorno' => ['required'],
143     ]);
144
145     $prestamo_update = PrestamoModel::find($id);
146     $prestamo_update->cliente_id = $request->get('cliente_id');
147     $prestamo_update->producto_id = $request->get('producto_id');
148     $prestamo_update->cantidad = $request->get('cantidad');
149     $prestamo_update->retorno = $request->get('retorno');
150     $prestamo_update->save();
151
152     $stock_producto = StockProductoModel::select('producto_id','cantidad')
153         ->where('producto_id',$prestamo_update->producto_id)->first();
154     $cantidad_restaurada = $cantidad_prestamo->cantidad + $stock_producto->cantidad;
155     $actualizar_cantidad_prestamo = $cantidad_restaurada - $prestamo_update->cantidad;
156
157     if($request->get('retorno')==0){
158         $actualizar_cantidad = StockProductoModel::where('producto_id',$cantidad_prestamo->producto_id)
159             ->update(['cantidad' => $actualizar_cantidad_prestamo]);
160     }else if($request->get('retorno')==1){
161         $actualizar_cantidad = StockProductoModel::where('producto_id',$cantidad_prestamo->producto_id)
162             ->update(['cantidad' => $cantidad_restaurada]);
163     }
164
165     return redirect('/prestamo')->with('success', 'Prestamo actualizado!');
166 }

```

Figura N°177. Prestamos - Eliminar

```

174 public function destroy($id)
175 {
176     $cantidad_prestamo = PrestamoModel::select('cantidad','producto_id','retorno')->where('id',$id)->first();
177     $stock_producto = StockProductoModel::select('cantidad')
178         ->where('producto_id',$cantidad_prestamo->cantidad)->first();
179     $cantidad_restaurada = $stock_producto->cantidad + $cantidad_prestamo->cantidad;
180
181     if($cantidad_prestamo->retorno == 0){
182         $actualizar_cantidad = StockProductoModel::where('producto_id',$cantidad_prestamo->producto_id)
183             ->update(['cantidad' => $cantidad_restaurada]);
184     }
185
186     $prestamo_eliminar = PrestamoModel::find($id);
187     $prestamo_eliminar->delete();
188
189     return redirect('/prestamo')->with('success', 'Prestamo eliminado!');
190 }

```

Integración del código a la interfaz del módulo.

Figura N°178. Prestamos - Interfaz

```

1 @extends('inicio')
2
3 @section('main')
4
5 <div class="row">
6     <h2>Prestamo de productos</h2>
7     <div class="col-sm-12">
8         <table id="tabla_general" class="table table-striped">
9             <thead>
10                <tr>
11                    <th>#</th>
12                    <th>Cliente</th>
13                    <th>Producto</th>
14                    <th>Cantidad</th>
15                    <th>Fecha prestamo</th>
16                    <th>Devolución</th>
17                    <th>Fecha devolución</th>
18                    <th>Acciones</th>
19                </tr>
20            </thead>
21            <tbody>
22                <@php $i=1 ?>
23                @foreach($prestamos as $pre)
24                    <tr>
25                        <td><span>{{ $i++ }}</span></td>
26                        <td><span>{{ $pre->cliente }} ({{ $pre->apellido_paterno }} {{ $pre->apellido_materno }})</span></td>
27                        <td><span>{{ $pre->producto }}</span></td>
28                        <td><span>{{ $pre->cantidad }}</span></td>
29                        <td><span>{{ \Carbon\Carbon::parse($pre->fecha_prestamo)->formatLocalized('%Y-%m-%d') }}</span></td>
30                        <td><span>{{ $pre->retorno == 0 ?>
31                            No </span></td>
32                        @else
33                            Si </span></td>
34                        @endif
35                        <td><span>{{ $pre->retorno == 0 ?>
36                            <td><span>---</span></td>
37                        @else
38                            <td><span>{{ \Carbon\Carbon::parse($pre->fecha_devolucion)->formatLocalized('%Y-%m-%d') }}</span></td>
39                        @endif
40                    </tr>
41                @endforeach
42                <div class="botones">
43                    <a class="accion_editar" data-bs-toggle="tooltip" title="Editar" href="{{ route('prestamo.edit',$pre->id) }}" ><i class="fas fa-pen-square"></i></a>
44                    <span id="{{ $pre->id }}" action="{{ route('prestamo.destroy', $pre->id) }}" method="post">
45                        @csrf
46                        @method('DELETE')
47                        <button type="button" id="{{ $pre->id }}" class="accion_eliminar" onclick="eliminarRegistro(event)"><i class="fas fa-trash-alt"></i></button>
48                    </form>

```

## Comprobación de los campos de texto para el CRUD.

Figura N°179. Prestamos – Ingreso de datos

```
1 @extends('inicio')
2
3 @section('main')
4
5 <div class="container">
6   <div class="row justify-content-center">
7     <div class="col-md-8">
8       <div class="card">
9         <div class="card-header">
10          {{ __('Register') }}
11        </div>
12
13        <div class="card-body">
14          <form method="POST" action="{{ route('prestamo.store') }}">
15            @csrf
16
17            <div class="form-group row">
18              <label for="cliente_id" class="col-md-4 col-form-label text-md-right">{{ __('Cliente') }}</label>
19
20              <div class="col-md-6">
21                <select class="form-control @error('cliente_id') is-invalid @enderror" name="cliente_id" required>
22                  <option value="{ null }">-- Seleccionar --</option>
23                  @foreach ($clientes as $cl)
24                    <option value="{{ $cl->id }}">{{ $cl->nombre }} {{ $cl->apellido_paterno }} {{ $cl->apellido_materno }}</option>
25                  @endforeach
26                </select>
27
28                @error('cliente_id')
29                  <span class="invalid-feedback" role="alert">
30                    <strong>{{ $message }}</strong>
31                  </span>
32                @enderror
33              </div>
34            </div>
35
36            <div class="form-group row">
37              <label for="producto_id" class="col-md-4 col-form-label text-md-right">{{ __('Producto') }}</label>
38
39              <div class="col-md-6">
40                <select class="form-control @error('producto_id') is-invalid @enderror" name="producto_id" required>
41                  <option value="{ null }">-- Seleccionar --</option>
42                  @foreach ($productos as $pro)
43                    <option value="{{ $pro->id }}">{{ $pro->nombre }}</option>
44                  @endforeach
45                </select>
46
47                @error('producto_id')
48                  <span class="invalid-feedback" role="alert">
49                    <strong>{{ $message }}</strong>
50                  </span>
51                @enderror
52              </div>
53            </div>
54          </form>
55        </div>
56      </div>
57    </div>
58  </div>
59 </div>
```

### Acta de entrega del Sprint 4

Siendo las 18:30 horas del día 11 de abril del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura a los requerimientos realizados y encargados del Sprint 4, asimismo muestra las interfaces y el código realizado hasta el momento.

Verificadas las explicaciones y el sustento presentado por el señor Romero Rodríguez, Billy Jerson para la aprobación del Sprint 4, se decide aprobar por parte de la gerente general Estupiñan Palma, Patricia el término del Sprint 4, del proyecto "Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C".



Firma

## Sprint 5

A continuación, se mostrarán los requerimientos funcionales que se realizarán para el sprint 5.

Tabla N°41: *Sprint 5*

#	Requerimientos Funcionales	N° Historia	Nombre Historia	Prioridad	Tiempo
1	Muestre gráficas de fácil comprensión.	H14	Gráficos de los reportes	Alta	5
2	Muestre un mensaje de confirmación sobre el cambio de dato.	H15	Confirmación de cambio de datos	Alta	3
3	No permita el ingreso de datos incorrectos en los diferentes campos de texto a rellenar.	H16	Verificación de datos	Alta	2
4	Muestre el stock del producto seleccionado para su salida.	H17	Verificación del stock	Alta	2
Total					12

### Planificación del Sprint 5

Siendo las 9:30 horas del día 12 de mayo del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura de los requerimientos del sistema.

Analizada la lista de requerimientos por el señor Romero Rodríguez Billy Jerson solventa algunas dudas sobre los requerimientos y se compromete a cumplir con los requisitos establecidos por el Sprint 5.

Los presentes imparten su aprobación a los requerimientos de acuerdo a lo establecido en la planificación del Sprint 5 indicando que la fecha de entrega de este Sprint sería el día 23 de mayo de 2021.



Firma

## Cronograma de tareas del Sprint 5

Tabla N°42: Cronograma del Sprint 5

<b>Sprint 5</b>	<b>Requerimiento F.</b>			
		12 días	12/05/2021	23/05/2021
<b>Historia N°14: Gráfico de los reportes</b>	<b>RF9</b>	5 días	12/05/2021	16/05/2021
Elaboración de procedimientos para procesar el rango de fechas		2 días	12/05/2021	13/05/2021
Comprobación de la existencia de datos.		2 días	14/05/2021	15/05/2021
Integración del código a la interfaz del módulo.		1 día	16/05/2021	16/05/2021
<b>Historia N°15: Confirmación de cambio de datos</b>		<b>RF10</b>	3 días	17/05/2021
Confirmación del procesamiento de los datos.	2 días		17/05/2021	18/05/2021
Integración del código con la interfaz.	1 día		19/05/2021	19/05/2021
<b>Historia N°16: Verificación de datos</b>	<b>RF11</b>	2 días	20/05/2021	21/05/2021
Comprobación del tipo de dato según el campo.		1 día	20/05/2021	20/05/2021
Integración a la interfaz mediante un mensaje.		1 día	21/05/2021	21/05/2021
<b>Historia N°17: Verificación de stock</b>	<b>RF12</b>	2 días	22/05/2021	23/05/2021
Comprobación del stock con la BD.		1 día	22/05/2021	22/05/2021
Integración de la validación de stock a la interfaz.		1 día	23/05/2021	23/05/2021

En la tabla anterior se muestra el cronograma del Sprint 5, en donde se muestra las fechas en las que se realizarán los requerimientos funcionales con respecto a cada una de sus historias de usuario.



## Historia N°14: Gráfico de los reportes

Elaboración de procedimientos para procesar el rango de fechas

Figura N°180. Reportes – Procesar fecha

```
45 public function store(Request $request)
46 {
47     $labels = collect();
48     $valores = collect();
49
50     $request->validate([
51         'start' => 'required',
52         'end' => 'required',
53     ]);
54
55     $cantidad_dia = DB::table('detalle_salidas as ds')
56         ->selectRaw('DATE_FORMAT(ds.created_at, "%d") as dia')
57         ->selectRaw('DATE_FORMAT(ds.created_at, "%M") as mes')
58         ->selectRaw('DATE_FORMAT(ds.created_at, "%a") as nombre_dia')
59         ->selectRaw('SUM(ds.cantidad) as cantidad_pro')
60         ->join('productos as pro', 'pro.id', '=', 'ds.producto_id')
61         ->join('salidas as s', 's.id', '=', 'ds.salida_id')
62         ->whereBetween(DB::raw('DATE_FORMAT(ds.created_at, "%Y-%m-%d")'), [$request->start, $request->end])
63         ->orderBy('ds.created_at', 'asc')
64         ->groupBy('dia')
65         ->whereNull('ds.deleted_at')
66         ->get();
67
68     \Carbon\Carbon::setUtf8(true);
69
70     foreach ($cantidad_dia as $value) {
71         $labels->push($value->dia . " " . (\Carbon\Carbon::parse($value->nombre_dia)->formatLocalized('%A')) . " " . (\Carbon\Carbon::parse($value->mes)->
72             formatLocalized('%B')));
73         $valores->push($value->cantidad_pro);
74     }
75
76
77     $chart = new SampleChart();
78     $chart->title("Salida de productos por día", 18, 'black', 'bold', 'Rockwell');
79     $chart->labels($labels);
80     $chart->dataset('Cantidad', 'line', $valores)
81         ->options([
82             'fill' => true,
83             'borderColor' => '#51C1C8']);
84 }
```

Comprobación de la existencia de datos.

Figura N°181. Reportes – Ingreso de la fecha

```
30 <div class="container">
31     <div class="row justify-content-center">
32         <div class="col-md-8">
33             <div class="card">
34                 <div class="card-header">{{ __('Reporte') }}</div>
35
36                 <div class="card-body">
37                     <form method="POST" action="{{ route('chart.store') }}">
38                         @csrf
39
40                         <div class="form-group row">
41                             <label for="start" class="col-md-4 col-form-label text-md-right">{{ __('Desde') }}</label>
42
43                             <div class="col-md-6">
44                                 <input id="start" type="date" class="form-control @error('start') is-invalid @enderror" name="start" value="" required
45                                     autocomplete="start" autofocus>
46
47                                 @error('start')
48                                     <span class="invalid-feedback" role="alert">
49                                         <strong>{{ $message }}</strong>
50                                     </span>
51                                 @enderror
52                             </div>
53                         </div>
54
55                         <div class="form-group row">
56                             <label for="end" class="col-md-4 col-form-label text-md-right">{{ __('Hasta') }}</label>
57
58                             <div class="col-md-6">
59                                 <input id="end" type="date" class="form-control @error('end') is-invalid @enderror" name="end" value="{{ old('precio') }}"
60                                     required autocomplete="end" autofocus>
61
62                                 @error('end')
63                                     <span class="invalid-feedback" role="alert">
64                                         <strong>{{ $message }}</strong>
65                                     </span>
66                                 @enderror
67                             </div>
68                         </div>
69
70                         <div class="form-group row mb-0">
71                             <div class="col-md-6 offset-md-4">
72                                 <button type="submit" class="btn btn-primary">
73                                     {{ __('Generar reporte') }}
74                                 </button>
75                             </div>
76                         </div>
77                     </form>
78                 </div>
79             </div>
80 </div>
81 </div>
```

Integración del código a la interfaz del módulo.

Figura N°182. Reportes – Interfaz

```
56 <div class="row pt-4">
57
58   <div style="width: 50%" class="col-md-6">
59     {!! $chart->container() !!}
60   </div>
61
62   {!! $chart->script() !!}
63   <div style="width: 50%" class="col-md-6">
64     {!! $pedidos->container() !!}
65   </div>
66
67   {!! $pedidos->script() !!}
68
69 </div>
```

## Historia N°15: Confirmación de cambio de datos

Confirmación del procesamiento de los datos.

Figura N°183. Confirmación de cambio de datos

```
136 $cliente_update = ClienteModel::find($id);
137 $cliente_update->nombre = $request->get('nombre');
138 $cliente_update->apellido_paterno = $request->get('apellido_paterno');
139 $cliente_update->apellido_materno = $request->get('apellido_materno');
140 $cliente_update->direccion = $request->get('direccion');
141 $cliente_update->referencia = $request->get('referencia');
142 $cliente_update->telefono = $request->get('telefono');
143 $cliente_update->email = $request->get('email');
144 $cliente_update->save();
145
146 return redirect('/cliente')->with('success', 'Cliente actualizado');
```

Integración del código con la interfaz.

Figura N°184. Mensaje de confirmación

```
58 @if (session('success'))
59   <div class="alert alert-success">
60     {{ session('success') }}
61   </div>
62 @endif
```

## Historia N°16: Verificación de datos

Comprobación del tipo de dato según el campo.

Figura N°185. Validación de datos

```
115 public function update(Request $request, $id)
116 {
117     $request->validate([
118         'nombre' => ['required', 'regex:/^[a-zA-Z\s-]+$/u'],
119         'apellido_paterno' => ['required', 'regex:/^[a-zA-Z\s-]+$/u'],
120         'apellido_materno' => ['required', 'regex:/^[a-zA-Z\s-]+$/u'],
121         'direccion' => ['required', 'string', 'max:255'],
122     ]);
123
124     if($request->telefono != null){
125         $request->validate([
126             'telefono' => ['bail', 'required', 'numeric', 'min:1000000', 'max:999999999'],
127         ]);
128     }
129
130     if($request->email != null){
131         $request->validate([
132             'email' => ['required', 'string', 'email', 'max:255'],
133         ]);
134     }
135 }
```

Integración a la interfaz mediante un mensaje.

Figura N°186. Validación de datos - Interfaz

```
1 @extends('inicio')
2 @section('main')
3
4 <div class="container">
5     <div class="row justify-content-center">
6         <div class="col-md-8">
7             <div class="card">
8                 <div class="card-header">{{ __('Editar' )}}</div>
9
10                 @if ($errors->any())
11                     <div class="alert alert-danger">
12                         <ul>
13                             @foreach ($errors->all() as $error)
14                                 <li>{{ $error }}</li>
15                             @endforeach
16                         </ul>
17                     </div>
18                     <br/>
19                 @endif
20
21                 <div class="card-body">
22                     <form method="post" action="{{ route('cliente.update', $cliente_edit->id) }}">
23                         @csrf
24                         @method('PATCH')
25
26                         <div class="form-group row">
27                             <label for="nombre" class="col-md-4 col-form-label text-md-right">{{ __('Nombres' )}}</label>
28
29                             <div class="col-md-6">
30                                 <input id="nombre" type="text" class="form-control @error('nombre') is-invalid @enderror" name="nombre" value="{{ $cliente_edit->nombre }}" required autocomplete="nombre" autofocus>
31
32                                 @error('nombre')
33                                     <span class="invalid-feedback" role="alert">
34                                         <strong>{{ $message }}</strong>
35                                     </span>
36                                 @enderror
37                             </div>
38                         </div>
39
40                         <div class="form-group row">
41                             <label for="apellido_paterno" class="col-md-4 col-form-label text-md-right">{{ __('Apellido Paterno' )}}</label>
42
43                             <div class="col-md-6">
44                                 <input id="apellido_paterno" type="text" class="form-control @error('apellido_paterno') is-invalid @enderror" name="apellido_paterno" value="{{ $cliente_edit->apellido_paterno }}" required autocomplete="apellido_paterno" autofocus>
45
46                                 @error('apellido_paterno')
47                                     <span class="invalid-feedback" role="alert">
48                                         <strong>{{ $message }}</strong>
49                                     </span>
50                                 @enderror
51                             </div>
52                         </div>
53                     </form>
54                 </div>
55             </div>
56         </div>
57     </div>
58 </div>
```

## Historia N°17: Verificación de stock

Comprobación del stock con la BD.

Figura N°187. Comprobación del stock con la BD

```
30 public function stock_producto(Request $request){
31     $stock_producto = StockProductoModel::select('cantidad')->where('producto_id',$request->get('id'))->first();
32
33     return response()->json($stock_producto->cantidad);
34 }
```

Integración de la validación de stock a la interfaz.

Figura N°188. Validación de stock - Script

```
75 $("#producto_id").change(function()
76 {
77     var id = $(this).find(":selected").val();
78     $.ajax
79     ({
80         url: '{{ route('stock') }}',
81         data: {
82             'id': id
83         },
84         dataType: 'json',
85         success: function(r)
86         {
87             $("#stock").val(r.val());
88         }
89     });
90 })
91
92 function verificarStock(event) {
93     event.preventDefault();
94
95     var form_id = event.currentTarget.id;
96     var cantidad = parseInt(document.getElementById('cantidad').value);
97     var stock = parseInt(document.getElementById('stock').value);
98
99     if(stock < cantidad){
100         Swal.fire({
101             icon: 'error',
102             title: 'Alerta',
103             text: 'La cantidad supera el stock del producto',
104         })
105     }else{
106         Swal.fire({
107             icon: 'success',
108             title: '¡Datos correctos!',
109             text: 'Registrando',
110             showConfirmButton: false,
111             timer: 1500
112         }).then(function() {
113             document.getElementById(form_id).submit();
114         });
115     }
116 }
117
118 </script>
```

### Acta de entrega del Sprint 5

Siendo las 19:30 horas del día 23 de mayo del 2021, se reúne mediante la plataforma zoom con la gerente general de la empresa Selva Gas Tocache S.A.C.

Presentes:

Nombres y Apellidos	Cargo
Patricia Estupiñan Palma	Gerente General
Billy Jerson Romero Rodríguez	Estudiante de la UCV

El señor Romero Rodríguez, Billy Jerson da lectura a los requerimientos realizados y encargados del Sprint 5, asimismo muestra las interfaces y el código realizado hasta el momento.

Verificadas las explicaciones y el sustento presentado por el señor Romero Rodríguez, Billy Jerson para la aprobación del Sprint 5, se decide aprobar por parte de la gerente general Estupiñan Palma, Patricia el término del Sprint 5, del proyecto "Desarrollo de un sistema web para el proceso de control de inventario en la empresa Selva Gas Tocache S.A.C".



Firma