



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**TESIS**

**Análisis comparativo de patrones de diseño MVC y  
MVP para el rendimiento de aplicaciones web**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS**

**Autor(es):**

**Bach. Gonzales Gonzales Christian Erick**

**ORCID: <https://orcid.org/0000-0002-5372-7073>**

**Asesor(a):**

**Mg. Cachay Maco Junior Eugenio**

**ORCID: <https://orcid.org/0000-0003-4056-3142>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú**

**2023**

**ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP  
PARA EL RENDIMIENTO DE APLICACIONES WEB**

**Aprobación Del Jurado**

---

MG. MEJÍA CABRERA HEBER IVAN  
PRESIDENTE DEL JURADO DE TESIS

---

MG. ATALAYA URRUTIA WILLIAN  
SECRETARIO DEL JURADO DE TESIS

---

MG. TUESTA MONTEZA VICTOR ALEXCI  
VOCAL DEL JURADO DE TESIS



## DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(n) la **DECLARACIÓN JURADA**, soy(somos) **egresado (s)** del Programa de Estudios de **Ingeniería de Sistemas** de la Universidad Señor de Sipán S.A.C, declaro(amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

### ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP PARA EL RENDIMIENTO DE APLICACIONES WEB

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Gonzales Gonzales Christian Erick	DNI: 73334419	
-----------------------------------	---------------	--

Pimentel, 07 de abril de 2023

## **Dedicatoria**

*Dirigido con mucho cariño a mis padres, por ese apoyo incondicional y esas ganas de verme por segunda vez profesional, por ser esa motivación extra para tener las fuerzas suficientes para realizarme como profesional y quienes son los pilares fundamentales que me brindan el apoyo en mi vida, con su infinito amor y sabios consejos que han sabido guiarme.*

*También una dedicatoria especial a mis docentes, por todas las enseñanzas recibidas durante toda la carrera universitaria.*

## **Agradecimiento**

*En primer lugar, a nuestro Dios padre que me brindó la oportunidad de cumplir uno de mis objetivos, por enseñarme el camino idóneo en mi vida, por darme las fuerzas para superar los obstáculos en mi carrera estudiantil, así también agradecer a todas las personas que estuvieron involucrados de manera indirecta y directa en la formación académica que obtuve en la universidad, a mis padres por brindarme el apoyo en toda etapa de mi vida, por ser un ejemplo de superación, esfuerzo y brindarme amor y comprensión en todo el trayecto de mi vida.*

## ÍNDICE

<b>Dedicatoria.....</b>	<b>iv</b>
<b>Agradecimiento.....</b>	<b>v</b>
<b>ÍNDICE DE FIGURAS .....</b>	<b>ix</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>xiii</b>
<b>Resumen.....</b>	<b>xvii</b>
<b>Abstract.....</b>	<b>xviii</b>
<b>I. INTRODUCCIÓN.....</b>	<b>19</b>
<b>1.1. Realidad Problemática .....</b>	<b>19</b>
1.1.1. Antecedentes de Estudio .....	20
<b>1.2. Formulación del problema.....</b>	<b>25</b>
<b>1.3. Hipótesis .....</b>	<b>28</b>
<b>1.4. Objetivos.....</b>	<b>28</b>
1.4.1. Objetivo General .....	28
1.4.2. Objetivo Especifico .....	28
<b>1.5. Teorías relacionadas al tema .....</b>	<b>28</b>
1.5.1. Software.....	28
1.5.2. Ingeniería de Software .....	29
1.5.2.1. Ingeniería Web.....	30
1.5.2.1.1. Aplicaciones Web.....	30
1.5.2.2. Desarrollo de Software .....	30
1.5.2.3. Patrones de Diseño .....	31
1.5.2.4. Tipos de Patrones de Diseño.....	35
1.5.2.4.1. Patrones de diseño orientado a objetos.....	35
1.5.2.4.2. Patrones de diseño de arquitectura .....	36
<b>II. MÉTODO .....</b>	<b>46</b>
<b>2.1. Tipo y Diseño de Investigación.....</b>	<b>46</b>
2.1.1. Tipo de Investigación: Tecnológica Aplicada.....	46
2.1.2. Diseño de la Investigación: Cuasi Experimental .....	46

<b>2.2.</b>	<b>Variables, Operacionalización.....</b>	<b>46</b>
2.2.1.	Operacionalización .....	47
2.2.2.	Variables.....	50
<b>2.3.</b>	<b>Población de estudio, muestra, muestreo y criterios de seleccion .....</b>	<b>50</b>
2.3.1.	Población .....	50
2.3.2.	Muestra .....	51
2.3.3.	Muestreo.....	51
2.3.4.	Criterios de Selección.....	52
<b>2.4.</b>	<b>Técnicas e instrumentos de recolección de datos, validez y confiabilidad</b>	<b>52</b>
2.4.1.	Técnica .....	52
2.4.2.	Instrumentos .....	52
<b>2.5.</b>	<b>Procedimientos de análisis de datos .....</b>	<b>53</b>
2.5.1.	Variable Dependiente .....	53
<b>2.6.</b>	<b>Criterios éticos.....</b>	<b>57</b>
2.6.1.	El consentimiento informado .....	57
2.6.2.	La confidencialidad .....	57
2.6.3.	Manejo de Riesgos .....	57
2.6.4.	Criterios de rigor científico .....	57
<b>III.</b>	<b>RESULTADOS.....</b>	<b>59</b>
<b>3.1.</b>	<b>Resultados.....</b>	<b>59</b>
3.1.1.	Determinación de la variable desarrollo .....	59
3.1.1.1.	Tiempo de Desarrollo.....	59
3.1.1.2.	Líneas de código escritas.....	74
3.1.2.	Interpretación de los resultados de la variable desarrollo .....	88
3.1.3.	Determinación de la variable recursos .....	88
3.1.3.1.	Uso de memoria RAM.....	90
3.1.3.2.	Uso de CPU.....	108
3.1.3.3.	Tiempo de respuesta.....	126
3.1.4.	Interpretación de los resultados de la variable Recursos .....	136
3.1.5.	Comparación porcentual de los patrones de diseño según sus variables	137
<b>3.2.</b>	<b>Discusión.....</b>	<b>138</b>
3.2.1.	Discusión con otras propuestas.....	138

3.2.2.	Discusión de los resultados de la presente investigación .....	140
<b>3.3.</b>	<b>Aporte de la investigación .....</b>	<b>143</b>
3.3.1.	Aporte Practico .....	143
3.3.2.	Seleccionar dos Patrones de Diseño.....	145
3.3.3.	Desarrollar una aplicación utilizando los dos patrones de diseños.....	157
3.3.3.1.	Selección del caso de estudio .....	157
3.3.3.2.	Selección de la metodología ágil de desarrollo de software.....	162
3.3.3.3.	Selección del lenguaje de programación .....	166
3.3.3.4.	Desarrollo de la aplicación utilizando la metodología XP.....	169
<b>IV.</b>	<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>215</b>
4.1	Conclusiones .....	215
4.2	Recomendaciones .....	217
	<b>REFERENCIAS.....</b>	<b>218</b>
	<b>ANEXOS.....</b>	<b>221</b>

## ÍNDICE DE FIGURAS

Figura 1: Patrones de Diseño orientado a objetos Gof. Fuente: [14] .....	35
Figura 2: Esquema del patrón de diseño Modelo, Vista y Presentador (MVP). Fuente: [27]38	
Figura 3: Esquema del patrón de diseño Modelo, Vista y Controlador (MVC). Fuente: [26]40	
Figura 4: Esquema del patrón de diseño Precontrolador, Controlador, Modelo, Vista y Postcontrolador (MVC Extendible). Fuente: [1] .....	42
Figura 5: Esquema del patrón de diseño Modelo, Vista y Vista Modelo (MVVM). Fuente: [4] .....	44
Figura 6: Esquema del patrón de diseño Presentación, Abstracción y Presentación (PAC). Fuente: [17].....	44
Figura 7: Interpretación de la Variable Desarrollo. Fuente: <i>Elaboración propia</i> .....	88
Figura 8: Memoria RAM disponible para la utilización de los patrones de diseño. Fuente: <i>Elaboración propia</i> .....	90
Figura 9: Utilización de Memoria RAM al ejecutar la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	91
Figura 10: Utilización de Memoria RAM al ejecutar el proceso de Pedidos Web de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	91
Figura 11: Utilización de Memoria RAM al ejecutar el proceso de Reservasiones web de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	92
Figura 12: Utilización de Memoria RAM al ejecutar el proceso de Contactar a la pollería de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	92
Figura 13: Utilización de Memoria RAM al ejecutar el proceso de Registrar Producto de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	93
Figura 14: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	93
Figura 15: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	94
Figura 16: Utilización de Memoria RAM al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVC. Fuente: <i>Elaboración propia</i> .....	94
Figura 17: Utilización de Memoria RAM al ejecutar la aplicación web MVP. Fuente: <i>Elaboración propia</i> .....	95
Figura 18: Utilización de Memoria RAM al ejecutar el proceso de Pedidos Web de la aplicación web MVP. Fuente: <i>Elaboración propia</i> .....	96
Figura 19: Utilización de Memoria RAM al ejecutar el proceso de Reservasiones web de la aplicación web MVP. Fuente: <i>Elaboración propia</i> .....	96

Figura 20: Utilización de Memoria RAM al ejecutar el proceso de Contactar a la pollería de la aplicación web MVP. Fuente: Elaboración propia .....	97
Figura 21: Utilización de Memoria RAM al ejecutar el proceso de Registrar Producto de la aplicación web MVP. Fuente: Elaboración propia .....	97
Figura 22: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVP. Fuente: Elaboración propia.....	98
Figura 23: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVP. Fuente: Elaboración propia .....	98
Figura 24: Utilización de Memoria RAM al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVP. Fuente: Elaboración propia.....	99
Figura 25: Uso de CPU disponible para la utilización de los patrones de diseño. Fuente: Elaboración propia .....	109
Figura 26: Utilización de Uso de CPU al ejecutar la aplicación web MVC. Fuente: Elaboración propia.....	109
Figura 27: Utilización de Uso de CPU al ejecutar el proceso de Pedidos Web de la aplicación web MVC. Fuente: Elaboración propia.....	110
Figura 28: Utilización de Uso de CPU al ejecutar el proceso de Reservasiones web de la aplicación web MVC. Fuente: Elaboración propia .....	110
Figura 29: Utilización de Uso de CPU al ejecutar el proceso de Contactar a la pollería de la aplicación web MVC. Fuente: Elaboración propia .....	111
Figura 30: Utilización de Uso de CPU al ejecutar el proceso de Registrar Producto de la aplicación web MVC. Fuente: Elaboración propia .....	111
Figura 31: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVC. Fuente: Elaboración propia.....	112
Figura 32: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVC. Fuente: Elaboración propia .....	112
Figura 33: Utilización de Uso de CPU al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVC. Fuente: Elaboración propia.....	113
Figura 34: Utilización de Uso de CPU al ejecutar la aplicación web MVP. Fuente: Elaboración propia.....	114
Figura 35: Utilización de Uso de CPU al ejecutar el proceso de Pedidos Web de la aplicación web MVP. Fuente: Elaboración propia .....	114
Figura 36: Utilización de Uso de CPU al ejecutar el proceso de Reservasiones web de la aplicación web MVP. Fuente: Elaboración propia .....	115
Figura 37: Utilización de Uso de CPU al ejecutar el proceso de Contactar a la pollería de la aplicación web MVP. Fuente: Elaboración propia .....	115

Figura 38: Utilización de Uso de CPU al ejecutar el proceso de Registrar Producto de la aplicación web MVP. Fuente: Elaboración propia .....	116
Figura 39: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVP. Fuente: Elaboración propia.....	116
Figura 40: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVP. Fuente: Elaboración propia .....	117
Figura 41: Utilización de Uso de CPU al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVP. Fuente: Elaboración propia.....	117
Figura 42: Características de la computadora para las pruebas del indicador Tiempo de Respuesta. Fuente: Elaboración propia .....	126
Figura 43: Tiempo de Respuesta al realizar el proceso de Pedidos Web de la aplicación web MVC. Fuente: Elaboración propia .....	127
Figura 44: Tiempo de Respuesta al realizar el proceso de Reservaciones Web de la aplicación web MVC. Fuente: Elaboración propia.....	128
Figura 45: Tiempo de Respuesta al realizar el proceso de Contactar a la pollería de la aplicación web MVC. Fuente: Elaboración propia .....	128
Figura 46: Tiempo de Respuesta al realizar el proceso de Registrar Producto de la aplicación web MVC. Fuente: Elaboración propia.....	129
Figura 47: Tiempo de Respuesta al realizar el proceso de Registrar Venta de la aplicación web MVC. Fuente: Elaboración propia.....	129
Figura 48: Tiempo de Respuesta al realizar el proceso de Registrar Venta por Mesa de la aplicación web MVC. Fuente: Elaboración propia .....	130
Figura 49: Tiempo de Respuesta al realizar el proceso de Listar los Pedidos Web de la aplicación web MVC. Fuente: Elaboración propia .....	130
Figura 50: Tiempo de Respuesta al realizar el proceso de Pedidos Web de la aplicación web MVP. Fuente: Elaboración propia.....	131
Figura 51: Tiempo de Respuesta al realizar el proceso de Reservaciones Web de la aplicación web MVP. Fuente: Elaboración propia .....	132
Figura 52: Tiempo de Respuesta al realizar el proceso de Contactar a la pollería de la aplicación web MVP. Fuente: Elaboración propia .....	132
Figura 53: Tiempo de Respuesta al realizar el proceso de Registrar Producto de la aplicación web MVP. Fuente: Elaboración propia .....	133
Figura 54: Tiempo de Respuesta al realizar el proceso de Registrar Venta de la aplicación web MVP. Fuente: Elaboración propia .....	133
Figura 55: Tiempo de Respuesta al realizar el proceso de Registrar Venta por Mesa de la aplicación web MVP. Fuente: Elaboración propia .....	134

Figura 56: Tiempo de Respuesta al realizar el proceso de Listar los Pedidos Web de la aplicación web MVP. Fuente: Elaboración propia .....	134
<i>Figura 57: Interpretación de la Variable Recursos. Fuente: Elaboración propia .....</i>	<i>136</i>
Figura 58: Flujo del proceso que se sigue en la presente investigación. Fuente: Elaboración propia.....	144
Figura 59: Fases de la Metodología Extreme Programing (XP). Fuente: (Pressman, 2010) .....	166
Figura 60. Modelo Entidad Relación. Fuente: Elaboración propia .....	193
Figura 61. Modelo Relacional. Fuente: Elaboración propia .....	194
Figura 62. Base de Datos. Fuente: Elaboración propia. ....	195
Figura 63. Interfaz de Página de Inicio. Fuente: Elaboración propia.....	196
Figura 64. Acceso al Sistema. Fuente: Elaboración propia .....	196
Figura 65. Interfaz de la sección Clientes. Fuente: Elaboración propia .....	197
Figura 66. Registro de los Clientes. Fuente: Elaboración propia .....	197
Figura 67. Interfaz de la sección de productos. Fuente: Elaboración propia.....	198
Figura 68. Registro de los productos. Fuente: Elaboración propia .....	198
Figura 69. Interfaz de la sección de categoría de productos. Fuente: Elaboración propia	199
Figura 70. Registro de la categoría de productos. Fuente: Elaboración propia .....	199
Figura 71. Interfaz de la sección de mesas. Fuente: Elaboración propia.....	200
Figura 72. Registro de las Mesas. Fuente: Elaboración propia .....	200
Figura 73. Interfaz de reservaciones de mesas vía web. Fuente: Elaboración propia ...	201
Figura 74. Visualización y Registro de reservaciones de mesas vía web. Fuente: Elaboración propia.....	202
Figura 75. Interfaz de la sección de ventas directas y ventas por mesa. Fuente: Elaboración propia.....	202
Figura 76. Registro de las ventas directas. Fuente: Elaboración propia .....	203
Figura 77. Registro de las ventas por mesa. Fuente: Elaboración propia.....	203
Figura 78. Interfaz de pedidos online. Fuente: Elaboración propia.....	204
Figura 79. Interfaz del pedido de un producto. Fuente: Elaboración propia.....	204
Figura 80. Interfaz de los registrados de pedidos online. Fuente: Elaboración propia ...	205
Figura 81. Interfaz de contactar. Fuente: Elaboración propia .....	205
Figura 82. Interfaz de contactar. Fuente: Elaboración propia .....	206

## ÍNDICE DE TABLAS

Tabla 1.		
Patrones de diseño de software .....		50
Tabla 2.	Patrones de	
diseño de software seleccionados.....		51
Tabla 3.		
Operacionalización de variables .....		47
Tabla 4.		
Descripción del Indicador Tiempo de Desarrollo .....		53
Tabla 5.		
Descripción del Indicador Líneas de Código Escritas.....		54
Tabla 6.		
Descripción del Indicador Uso de Memoria RAM .....		55
Tabla 7.	Descripción	
del Indicador Uso de CPU.....		55
Tabla 8.	Descripción del	
Indicador Tiempo de Respuesta .....		56
Tabla 9.		
Bitácora de Tiempo de Desarrollo del Patrón de Diseño MVC .....		60
Tabla 10.		
Bitácora del Patrón de Diseño MVP .....		67
Tabla 11:	Resultados	
obtenidos respecto al Tiempo de Desarrollo .....		73
Tabla 12.		
Bitácora de Líneas de Código Escritas del Patrón de Diseño MVC.....		75
Tabla 13:		
Bitácora de Líneas de Código Escritas del Patrón de Diseño MVP.....		82
Tabla 14:	Resultados	
obtenidos respecto al indicador Líneas de Código Escritas.....		87
Tabla 15:	Escenario para	
realizar las pruebas de la Variable Recursos .....		89
Tabla 16.	Resultados	
obtenidos respecto al indicador de Memoria Ram.....		100
Tabla 17:	Resultados	
obtenidos respecto al indicador de uso de CPU.....		118

Tabla 18:	Resultados	
obtenidos respecto al indicador Tiempo de Respuesta .....		135
Tabla 19.		
Descripción y Análisis de los Patrones de Diseño .....		146
Tabla 20.		
Cantidad de Aplicaciones Web a desarrollar .....		152
Tabla 21.		
Selección de los Patrones de Diseño .....		154
Tabla 22.		
Selección de Caso de Estudio .....		158
Tabla 23.		
Selección de la metodología ágil de desarrollo de software .....		163
Tabla 24.		
Selección del lenguaje de programación.....		167
Tabla 25.	Roles del	
Equipo del Proyecto .....		171
Tabla 26.		
Historia de Usuario 1: Página de bienvenida y Acceso al sistema.....		174
Tabla 27.		
Historia de Usuario 2: Usuarios y validación de usuarios .....		175
Tabla 28.		
Historia de Usuario 3: Permisos de Usuario .....		176
Tabla 29.		
Historia de Usuario 4: Registro y lista de usuarios .....		177
Tabla 30.		
Historia de Usuario 5: Sección de clientes .....		178
Tabla 31.		
Historia de Usuario 6: Registro de Clientes .....		178
Tabla 32.		
Historia de Usuario 7: Sección de Productos .....		179
Tabla 33:		
Historia de Usuario 8: Registro de Productos.....		179
Tabla 34.		
Historia de Usuario 9: Sección de Categorías de Productos .....		180
Tabla 35.		
Historia de Usuario 10: Registro de Categorías de Productos.....		180

Tabla 36.	
Historia de Usuario 11: Sección de Mesas.....	181
Tabla 37.	
Historia de Usuario 12: Registro de mesas .....	181
Tabla 38.	
Historia de Usuario 13: Sección de reservaciones de mesa vía web.....	182
Tabla 39.	Historia
de Usuario 14: Registro de reservaciones de mesa vía web .....	182
Tabla 40.	
Historia de Usuario 15: Sección de ventas directa y ventas por mesa .....	183
Tabla 41.	Historia
de Usuario 16: Registro de las ventas directas .....	183
Tabla 42.	Historia
de Usuario 17: Registro de las ventas por mesa .....	184
Tabla 43.	Historia
de Usuario 18: Sección de pedidos online .....	184
Tabla 44:	Historia
de Usuario 19: Registro de Pedidos Online.....	185
Tabla 45.	Historia
de Usuario 20: Sección de Contactar.....	185
Tabla 46:	
Historia de Usuario 21: Registro de Contactar .....	186
Tabla 47.	Planificación de
los lanzamientos .....	187
Tabla 48.	Iteraciones
del Patrón de Diseño MVC.....	188
Tabla 49.	
Iteraciones del Patrón de Diseño MVP .....	190
Tabla 50.	
Prueba de aceptación: Pagina de bienvenida y Acceso al sistema .....	207
Tabla 51.	Prueba de
aceptación: interfaz y registro de clientes, interfaz y registro de productos, interfaz y registro de categoría de productos, interfaz y registro de mesas. ....	208
Tabla 52.	
Prueba de aceptación: Sección y Registro de reservaciones de mesa vía web .....	209

Tabla 53.	
Prueba de aceptación: Sección de ventas directa y ventas por mesa .....	210
Tabla 54.	
Prueba de aceptación: Registro de las ventas directas .....	211
Tabla 55.	Prueba de
aceptación: Registro de las ventas por mesa .....	212
Tabla 56.	
Prueba de aceptación: Sección y Registro de pedidos online .....	213
Tabla 57.	
Prueba de aceptación: Sección y Registro de contactar .....	214

## Resumen

Se realizó una evaluación empírica de los patrones de diseño en este trabajo de investigación, teniendo como objetivo medir su eficacia sobre los proyectos de desarrollo de software aplicado a las aplicaciones web. Uno de los problemas que ocurre comúnmente en los proyectos de software es la incertidumbre que se tiene para elegir los patrones de diseño adecuados para el diseño de un proyecto. Se inició con la identificación de los patrones de diseño existentes, para así someterlos a un proceso de validación, en el cual se asignaron puntuaciones de acuerdo al grado que solucionaban diferentes problemas de desarrollo de software.

Después de obtener los resultados del proceso, se establecieron 2 patrones de diseño para la evaluación. Así mismo, se realizó dos proyectos de desarrollo de software web que están diseñados con los dos patrones de diseño seleccionados en el proceso anterior.

Para cada software web se implementó cada patrón de diseño. Posteriormente a cada patrón de diseño se le evaluó la eficacia mediante los indicadores de tiempo de desarrollo, líneas de código escritas, uso de memoria RAM, uso de CPU y tiempo de respuesta. Para la medición de los indicadores se tomaron en cuenta el tiempo que utilizaron para implementar dichos patrones y un escenario de prueba que nos permite medir los recursos que consume en una computadora al hacer uso de patrones de diseño, además se seleccionó y aplicó una metodología ágil para desarrollar el software web.

Como resultado general se obtuvo que el patrón de diseño Modelo, Vista y Controlador (MVC), demostró una eficacia significativamente mayor al patrón de diseño Modelo, Vista y Presentador (MVP) en el desarrollo de software web.

### **Palabras claves:**

Ingeniería Web, proyecto de desarrollo de software web, metodología ágil, patrones de diseño orientado a objetos, patrones de diseño de arquitectura.

## **Abstract**

An empirical evaluation of the design patterns was carried out in this research work, with the objective of measuring their effectiveness on software development projects applied to web applications. One of the problems that commonly occurs in software projects is the uncertainty in choosing the right design patterns for the design of a project. It began with the identification of existing design patterns, in order to submit them to a validation process, in which scores were assigned according to the degree that they solved different software development problems.

After obtaining the results of the process, 2 design patterns were established for the evaluation. Likewise, two web software development projects were carried out that are designed with the two design patterns selected in the previous process.

For each web software each design pattern was implemented. Subsequently, each design pattern was evaluated for its effectiveness through development time indicators, lines of code written, RAM memory usage, CPU usage, and response time. For the measurement of the indicators, the time they used to implement said patterns and a test scenario that allows us to measure the resources consumed in a computer when making use of design patterns were taken into account, in addition, an agile methodology was selected and applied. to develop web software.

As a general result, it was obtained that the Model, View and Controller (MVC) design pattern demonstrated significantly greater efficiency than the Model, View and Presenter (MVP) design pattern in web software development.

### **KeyWords:**

Web engineering, web software development project, agile methodology, object-oriented design patterns, architecture design patterns.

## I. INTRODUCCIÓN

### 1.1. Realidad Problemática

Las webs ofrecían funcionalidades simples que mostraba descripciones e imágenes que brindaba la información, pero en la actualidad existen web que además de brindarte información también te brindan funcionalidades más complejas. Es ahí la importancia de tener un buen diseño al desarrollar una aplicación web.

En la actualidad el uso de patrones de diseño rápidamente se ha extendido de tal manera que es casi un estándar en el desarrollo de software ya que simplifica mediante una arquitectura el diseño de agregar funcionalidades al software sin que afecte su funcionalidad inicial. Además, nos brinda un producto de software confiable y útil, más rápido y con un mejor control.

Existe una variedad de patrones de diseño, por lo que dificulta seleccionar el apropiado y el más útil. No solo se puede seleccionar uno, sino que además se puede adaptar los patrones de diseño entre si y así brindar soluciones a problemas que se tiene al desarrollar el software.

Para la selección del patrón de diseño se presenta el problema de tener mucha información en artículos, libros y sitios web; conllevando a la confusión en los diseñadores para elegir el patrón de diseño conveniente.

Actualmente hay publicaciones de investigaciones que tienen relación con el tema en cuestión, en el que se muestra como el diseñador comete errores para seleccionar un patrón adecuado, conllevando un atraso de tiempo y perdidas económicas para utilizar un patrón de diseño conveniente.

### **1.1.1. Antecedentes de Estudio**

Los autores [1], en la investigación “Pre and Post Controller based MVC Architecture for Web Application”, desarrollado en 5th International Conference on Informatics, Electronics and Vision (ICIEV) en el año 2016; identifico el problema que el Patrón de Diseño MVC al ser implementado en aplicaciones web se vuelve más complejo y se necesita más tiempo de trabajo. En dicha investigación se realizó la propuesta del marco MVC extendido. Para realizar la comparación del MVC Extendible, Silez y Slim se hizo la evaluación utilizando las herramientas el Apache Benchmark que mide su rendimiento. Al final de la investigación se obtuvo que el MVC Extendible respecto a la solicitud por segundo y el tiempo por solicitud un resultado que durante 30 segundos con 30 niveles simultáneos propuestos es 115.52, 39.58 y 45.2. La relación con la investigación se ha dado al comparar el rendimiento de los patrones de diseño mediante la herramienta Benchmark.

Los autores [2], en la investigación “The Software Design of Modular Instrument Based on Proxy Pattern”, desarrollado en Chinese Automation Congress (CAC) en el año 2018; identifico el problema del instrumento modular debido a que su interfaz de desarrollo no es independiente. En dicha investigación se ha propuesto un nuevo diseño basado en el patrón proxy. Para realizar la propuesta comienza con el análisis de requisitos del módulo de conversión descendente de banda ancha y proporciona al módulo una arquitectura de diseño de software basada en el patrón proxy. Al final de la investigación obtuvimos que el nuevo marco es más eficiente ya que al empaquetar datos de 125 kilobytes, el tiempo de transmisión es de 170 ms en comparación al instrumento modular tradicional que es de 1,2 s; además de que al implementarlo se reduce líneas de código. La relación con la investigación se da

porque han propuesto y comparado un nuevo patrón de diseño basado en el patrón de diseño proxy para mejorar la eficiencia del instrumento modular.

Los autores [3], en la investigación “Design of Intelligent Drilling System Software Framework and Data Architecture Based on MVC Pattern”, desarrollado en Chinese Control Conference (CCC) del año 2019; identifico el problema de eficiencia de perforación en el software del sistema de perforación especialmente en el complejo entorno geológico. En dicha investigación propusieron un marco basado en el patrón MVC en el cual la parte del modelo lo divide en modelo principal y auxiliar. Para realizar la propuesta el modelo principal incluye la evaluación de seguridad, eficiencia, coordinación del proceso y la fusión de datos; y el modelo auxiliar incluye la sincronización de base de datos, gestión de usuarios y configuración de parámetros. Al final de la investigación se obtuvo que el nuevo marco es más eficiente ya que cada módulo llama directamente a los datos, permitiendo ser independientes y mejorando la escalabilidad del sistema. La relación con la investigación al proponer el nuevo patrón de diseño basado en el MVC para mejorar la eficiencia de la perforación.

Los autores [4], en la investigación “Using Design Patterns to Improve Web Service Performance”, desarrollado en IEEE International Conference on Services Computing en el año 2015, identifico el problema en el tiempo de respuesta y el rendimiento de los servicios web de búsqueda. En esta investigación se realizó la evaluación del enfoque tradicional y los patrones de diseño Singleton y el Patrón de Almacenamiento Caché mediante el modo secuencial y simultaneo. Para realizar la evaluación el modo secuencial consiste en solicitudes y el modo simultaneo proporcionando más dinamismo simulando solicitudes concurrentes de consumidores de servicios. Al final de la investigación se obtuvo que el patrón de Almacenamiento Cache brinda un rendimiento mejor en los dos modos, en el modo secuencial obteniendo como valores el enfoque tradicional, los patrones singleton y caching que son 0.39, 0.45 y 0.72

respectivamente, y en el modo simultaneo, en tiempo medio de respuesta de consumidores se obtiene que cada 50 respuestas es 90 y 280 respectivamente, en ahorro de tiempo de respuesta alcanza alrededor 70,5% y 50 % respectivamente, y en solicitudes por unidad de tiempo es 6,2 y 1,9. La relación con la investigación se da al comparar el rendimiento de los patrones de diseño mediante el modo secuencial y simultaneo.

Los autores [5], en la investigación “Design patterns for Machine Learning Applications”, desarrollado en 3rd International Conference on Computing Methodologies and Communication (ICCMC) en el año 2019; identifico el problema en dos aplicaciones que utilizan técnicas de Machine Learning (ML) y Deep Learning que no están diseñadas con patrones de diseño. En dicha investigación se propusieron detectar y analizar patrones de diseño y patrones de arquitectura para implementarlos en dichas aplicaciones. Para realizar el análisis se basa en los principios de diseño estándar y patrones arquitectónicos de los cuales se seleccionó y destaco el patrón de diseño Composite y el patrón de arquitectura MVC. Al final de la investigación se llegó a la conclusión que la gran variedad de patrones de diseño se puede utilizar para varios propósitos. La relación con la investigación se da el análisis de los patrones de diseño y así implementarlo en dos aplicaciones.

Los autores [6], en la investigación “Variadic Parameter Command Pattern and the Utilization in MVC”, desarrollado en 9th International Conference on Software Engineering and Service Science (ICSESS) en el año 2018; identifico el problema que el Patrón de Diseño MVC al tener una pobre universalidad. En dicha investigación se propuso el patrón que tiene más universalidad el cual inserta una capa de comando

entre el modelo y el controlador. Para realizar la propuesta agrega un comando de parámetro variádico que es una característica de la plantilla de C++ 11 el cual admite diferentes tipos de valor de retorno. Al final de la investigación obtuvimos que el patrón de diseño propuesto brinda más efectividad debido a que el comando puede ser heredado y ampliado lo que evita el trabajo redundante de diseñar diferentes clases de comando subconcretas para cada operación. La relación con la investigación se da porque han propuesto un nuevo patrón de diseño en el cual agregan o modifican componentes permitiendo tener mejores resultados.

Los autores [7], en la investigación “A Presentation Framework to Simplify the Development of Java EE Application Thin Clients”, presentada en Eighth International Conference on Complex, Intelligent and Software Intensive Systems en el año 2014, identifico el problema de utilizar el patrón MVC debido a las tareas que se repiten al realizar aplicaciones haciéndolo molesto y complicado. En dicha investigación propusieron un marco que admite el nivel de presentación. Para realizar la propuesta primeramente se realizó una revisión de los importantes marcos de software y patrones de diseño que tienen compatibilidad con Java EE, después se presentó el marco y al final se implementó, la evaluación se realizara con la aplicación PruebaDerbyUOC y lo ejecutaron con el servidor GlassFish. Al final de la investigación se obtuvo que la propuesta ofrece un óptimo rendimiento ya que facilita y simplifica mucho el desarrollo de las pruebas. La relación con la investigación se da al proponer un marco de software que brinda un eficiente rendimiento que el patrón MVC.

Los autores [8], en la investigación “A Feasibility Analysis for the Application of Design Patterns in Search Based Product Line Design”, desarrollado en 32nd International Conference of the Chilean Computer Science Society (SCCC) 2013, identifico el problema al no encontrar enfoques que buscan la aplicación de patrones de diseño en el software de Arquitecturas de línea de producto (PLA). En dicha investigación propusieron realizar un análisis de factibilidad a los patrones GoF (Gang of Four). Para realizar la evaluación se analizó la estructura, consecuencias del uso de acuerdo con métricas arquitectónicas, flexibilidad y aspectos de automatización. Al final de la investigación se obtuvo que Cuatro patrones (Bridge, Strategy, Facade y Mediator) son factibles y dos patrones (Bridge y Strategy), de ellos tienen ámbitos de aplicación Línea de Productos de Software (SPL) específicos que implican variabilidades. La relación con la investigación se da al realizar un análisis y selección de los patrones de diseño para así aplicarlos en las búsquedas.

Los autores [9], en su investigación “Design pattern evaluation of mobile and web based application frameworks”, desarrollada en Sixth International Conference on Developments in eSystems Engineering 2013, identifico el problema en técnicas para aplicaciones web de escritorio debido que no transmite suficientemente bien al entorno móvil. En dicha investigación propusieron el uso del método empírico debido al mecanismo eficaz y además el investigador no tiene control de las variables por lo que está centrado en un conjunto contemporáneo de los acontecimientos. Para realizar la propuesta primeramente se define y examina que patrones de diseño están incluidos y cuales son útiles para los marcos web (ASP.NET MVC, Struts, JSF, Grails y Zend) y el marco móvil (Android, iOS, MonoCross, Appcelerator Titanium, Sencha Touch, JQuery Mobile, AppMobi, PhoneGap), los patrones a consideración son MVC, Front Controller y Contextos y la Inyección de Dependencias (CDI) y luego se calificaron con un valor 0 y 1, donde 1 es la puntuación más alta. Al final de la investigación obtuvimos

al patrón MVC siendo el más utilizado en los marcos ya que el puntaje medio obtenido para el desarrollo móvil es de 0,59; 0,16 (Front Controller) y 0,24 (CDI). La relación con la investigación se da ya que se realiza un método empírico a los patrones de diseño, en el cual se demuestra que el mas utilizado es el patrón MVC.

Los autores [10], en la investigación “Application of MVVM Design Pattern in MES”, desarrollada en Cyber Technology in Automation, Control, and Intelligent Systems (CYBER) 2015, identifico el problema del Sistema de Ejecución de Manufactura (MES) que desarrolla requisitos personalizados para las empresas de fabricación. En dicha investigación se propone el patrón de diseño MVVM debido a la separación del negocio y la lógica, y aumenta la reutilización de código. Para realizar la evaluación analizaron el contexto, estructura y consecuencias, luego se desarrolla con .Net Frame Work y la utilización generalizada de WPF. Al final de la investigación obtuvimos como resultado que este patrón tiene atributos de alta cohesión y bajo acoplamiento; también hay reducción de código de desarrollo. La relación con la investigación está dada porque se implementó el patrón de diseño MVVM en el diseño MES obteniendo un buen rendimiento al realizar cambios de requerimiento del cliente.

## **1.2. Formulación del problema**

¿Qué Patrón de Diseño de software tiene un mejor rendimiento en las aplicaciones web?

### **– Justificación e importancia del estudio**

La presente investigación se justifica y por las siguientes razones es importante:

- **Pertinencia de la investigación**

Esta es una investigación pertinente porque es oportuna y conveniente. Es oportuna porque el tema de Patrones de Diseño es inherente cuando se analiza, diseña o implementa soluciones de desarrollo de software. En ese sentido los desarrolladores de software tienen en los patrones de diseño una guía que orienta las tareas que debe desarrollar cuando implementa los diferentes componentes que conforman el producto de software. Debe entenderse que el desarrollo de software es una de las actividades fundamentales de todo profesional en Ingeniería de Sistemas.

Es conveniente, porque al considerar un patrón de diseño se evita una serie de errores que normalmente ejecuta un desarrollador como: incoherencias en la integración de los componentes de producto del software, la imposibilidad de crecimiento del producto o la mantenibilidad del mismo, etc.

Esta investigación aportará con una guía práctica de la forma con la que se debe aplicar los dos patrones de diseño analizados.

- **Relevancia de la investigación**

En términos de relevancia, esta tesis pretende aportar, luego del análisis comparativo de los dos Patrones de Diseño estudiados, información empírica sobre la forma de cómo se aplica estos patrones en la distribución de componentes del software, en el diseño de los componentes y las ventajas y desventajas de cada uno de estos en el desarrollo de software.

El desarrollador es el beneficiario directo, porque lo ayudaremos a reutilizar patrones de diseño en el software que le permitirán minimizar los riesgos de generar un mal diseño.

- **Acotación de la investigación**

El interés al desarrollar esta investigación es profundizar los conocimientos adquiridos, en relación al desarrollo de software, de tal forma que permitan determinar cuándo es conveniente utilizar un patrón de referencia en particular. Los proyectos de software, si bien es cierto tratan de atender requerimiento y necesidades de información, la ingeniería de este implica que no siempre se utilice la misma arquitectura para la distribución de sus componentes. Cada producto de software tiene sus propias características de arquitectura.

Por lo tanto, esta investigación nos permitirá determinar con mayor certeza cuando aplicar un patrón de referencia determinado, empezando con los patrones de diseño MVC y MVP y que posteriormente puede ser extendido a otros.

- **Viabilidad de la investigación**

La investigación, desde toda perspectiva es factible; específicamente sustentado en su factibilidad operativa, tecnológica y económica.

Es factible operativamente porque los dos patrones de diseño que van a hacer evaluados son los más conocidos aplicados y documentados, por lo tanto, no existe dificultad para su comprensión.

Es factible tecnológicamente, porque para el análisis de los dos patrones de referencia estudiados no se requiere de una infraestructura que esta fuera de lo común para cualquier entorno o ambiente de desarrollo. Es aplicable en cualquier tipo de lenguaje de programación y motor de base de datos.

Y es factible económicamente porque las pruebas experimentales e realizaran en entornos o ambientes de software OpenSource.

### **1.3. Hipótesis**

El patrón de diseño de mejor rendimiento es el MVP (Modelo, Vista y Presentador) para el desarrollo de aplicaciones web, porque nos permite reducir tiempo significativos al desarrollar aplicaciones web.

### **1.4. Objetivos**

#### **1.4.1. Objetivo General**

Evaluar los patrones de diseño de software aplicado a las aplicaciones web.

#### **1.4.2. Objetivo Especifico**

- a) Seleccionar dos Patrones de Diseño de Software.
- b) Desarrollar dos aplicaciones web utilizando los dos Patrones de diseños de software seleccionados.
- c) Realizar las mediciones de acuerdo al rendimiento de los indicadores en las aplicaciones web.

### **1.5. Teorías relacionadas al tema**

#### **1.5.1. Software**

Son instrucciones detalladas para hacer una tarea asignada, y para hacer dicha tarea se necesita recibir el orden. [18]

Al interactuar el usuario y la computadora, se le determina como componente lógico, ya que por medio de periféricos de entrada el usuario envía órdenes y la computadora lo interpreta por medio de periféricos de salida. [19]

Nos describe que son efectuadas los programas en las maquinas, y nos dice además que es un medio que brinda un servicio para realizar difíciles funcionalidades. Al elaborar software tiene como concepto una actividad económica debido que brinda una gran ayuda a varias personas. [20]

### **1.5.2. Ingeniería de Software**

Abarca dos aspectos en la producción del software, el primero engloba herramientas y teorías ya que solucionan los problemas que se localizan y en el segundo están teorías de apoyo, gestión de proyectos y desarrollo de métodos y herramientas. En dichos aspectos están basados al producir software que empiezan en primeras etapas hasta mantenerlo luego que se utilizó el software. [21]

Para ofrecer calidad en el software, se tiene usabilidad, eficacia y mucho más. Además, de funcionar el software también tiene que tener calidad, y para lograrlo se usa una serie de métodos, técnicas y más. No solamente comprende una fase de desarrollo, sino que engloba a las fases. [22]

### **1.5.2.1. Ingeniería Web**

#### **1.5.2.1.1. Aplicaciones Web**

[23]. Los navegadores web soportan estas aplicaciones y son codificados por un lenguaje de programación, permite la interacción entre el usuario y servidor.

[24]. Está basado en dos lados su arquitectura que son el lado cliente, que hace uso de la aplicación en los navegadores web (Google Chrome, Firefox), y realizar la interacción con la aplicación y el otro lado llamado servidor, que se encarga de residir los datos, reglas y la lógica de la aplicación. Para tener acceso en las aplicaciones lo haces mediante la computadora y otros dispositivos, lo cual no nos limita el lugar en el cual está ubicado el usuario ya que solo tiene que tener internet.

### **1.5.2.2. Desarrollo de Software**

Según [25], el ofrecer un software de calidad se tiene diferentes modelos de desarrollo, y tenemos etapas al desarrollarlo, los cuales son:

#### **a) Análisis**

Para la recopilación de requisitos de software, se utilizan técnicas que están basados en recopilar, examinar y formular el requerimiento del cliente.

#### **b) Diseño**

Está dividido en dos etapas, la fase de diseño general que asume los requerimientos generales de la arquitectura de la aplicación.

#### **c) Codificación**

Después de haber definido en la fase anterior las funciones, se realiza implementación del software a través del lenguaje de programación.

#### **d) Pruebas**

Se realizan en la fase unidad es decir pruebas individuales y la prueba de integridad el cual asegura la integración de diferentes módulos junto a la aplicación.

#### **e) Mantenimiento**

En esta fase hay actualizaciones secundarias y correcciones del software.

En conclusión, para brindar un software de calidad es importante cada fase de desarrollo, una de esas fases es el de diseño, en el cual se habla de la arquitectura, es ahí la importancia de los Patrones de Diseño porque se aplican para obtener una arquitectura adecuada.

#### **1.5.2.3. Patrones de Diseño**

Según [12], nos dice que no son código estos patrones, ya que es una idea abstracta que brinda una plantilla que al realizar una adaptación con el desarrollo del software nos soluciona un problema determinado que se tiene en el diseño.

Tiene que cumplir con 2 requerimientos indispensables para que sea considerado patrón:

- a) El primer requerimiento es la solución que resuelva el problema tiene que ser muy efectiva dicha solución y la mejor elección al ser comparado con otras posibles que se tiene.
- b) El segundo es que al tener un problema se reutilice la solución.

El propósito de utilizarlos

- a) Para el desarrollo de software se tiene un grupo de pares problema-solución con la finalidad de poder utilizarlos.
- b) Si está comprobado su efectividad se evitar una búsqueda de solución.
- c) Se impone estándares para realizar el diseño del software.

Los autores [13], nos dice que son capaces de brindar soluciones reutilizables para apoyar a las personas que no cuentan con experiencia.

Para la utilización de los patrones es no separarlos porque al combinarlos nos solucionan problemas más complejos. Los podemos encontrar en catálogos y lenguajes de patrones,

- a. Se clasifican mediante el uso de criterios y así sea más rápido buscar una solución ante un problema. Ser un patrón lo primero es conocer el problema que solucionaremos, después verificar el catálogo, luego elegir que patrones se adecua mejor a lo que se necesita.

- b. El lenguaje de patrones tiene más complejidad su clasificación debido a la aplicación de un grupo de patrones y en el orden que se aplicaron ayudándonos a tener más efectividad en el proceso.

La estructura del patrón de diseño es:

**1) Nombre**

Tiene que ser único el nombre ya que nos va a permitir la identificación de cada patrón.

**2) Problema**

En una o dos frases describe los objetivos que alcanza cada patrón.

**3) Contexto**

A un problema recurrente se aplica el patrón.

**4) Fuerzas**

Específica de cada patrón sus objetivos, restricciones relevantes y fuerzas, también la comunicación entre ellas o con las metas a conseguir.

**5) Solución**

Explica mediante instrucciones la construcción del producto resultante, esta explicación también se puede acompañar de diagramas de solución.

## **6) Ejemplo**

Pueden ser visuales, ayudando al lector a una mejor aplicación y comprensión del patrón.

## **7) Contexto resultante**

Después de haber aplicado el patrón, se señala el estado del sistema, que implica sus consecuencias negativas o positivas.

## **8) Exposición razonada**

Describe la funcionalidad y la utilidad del patrón, describiendo sus mecanismos subyacentes.

## **9) Patrones relacionados**

Estos patrones pueden agruparse entre sí o posiblemente ser aplicados partiendo del contexto resultante.

Al utilizar los patrones de diseño se tiene muchas ventajas, pero también hay desventajas debido a que hay limitaciones.

- a) Se tiene variedad de patrones de diseño publicados en artículos y libros, lo que conlleva una confusión de la comunidad al cual se dirige, provocando una serie de problemas; permitiendo que los lectores lleven a cabo una búsqueda manual para la identificación del patrón.
- b) Ya estando ubicado el patrón, el obstáculo posterior es su aplicación, debido a que sino comprendes a profundidad el problema que resuelve, no sabrás aplicarlo en su momento.

- c) Es posible utilizar otros patrones para concluir una solución, conllevando la repetición de pasos anteriores.
- d) Si no está soportado metodológicamente se tendría otro problema el uso de los patrones, salvo en algún dominio concreto.
- e) Otra de las dificultades para los diseñadores es localizar patrones que tengan nombres diferentes pero que abordan el mismo problema del autor.

#### 1.5.2.4. Tipos de Patrones de Diseño

##### 1.5.2.4.1. Patrones de diseño orientado a objetos

Según [14], estos patrones los organiza en base a dos criterios: propósito (de creación, estructurales y de comportamientos) y ámbito (clase y objeto).

		Propósito		
		De Creación	Estructurales	De Comportamiento
Ámbito	Clase	Factory Method	Adpater (de clases)	Interpreter Template Method
	Objeto	Abstract Factory Builder Prototype Singleton	Adapter (de objetos) Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Figura 1: Patrones de Diseño orientado a objetos Gof. Fuente: [14]

### **A. Patrones de creación**

Apoyan al sistema para tener independencia de cómo está representado, creado, compuesto sus objetos. Utilizan la herencia para modificar de la instancia para la creación.

### **B. Patrones Estructurales**

Forman estructuras grandes a través de la unión de clases y objetos. La herencia es utilizada para componer interfaces o implementaciones.

### **C. Patrones de Comportamiento**

Son algoritmos y se asigna responsabilidades de objetos.

#### **1.5.2.4.2. Patrones de diseño de arquitectura**

Según el autor [26], nos dice que la arquitectura de software esta organizados por componentes del sistema, que se encuentran están relacionados con el contexto y entre sí mismos.

##### **1) Patrón de diseño Modelo, Vista y Presentador (MVP)**

El autor [15], dice la estructura del patrón tiene más flexibilidad al implementarse una función en la clase especificada, además tiene la facilidad de realizar varios escenarios de prueba mejorando la capacidad de mantenimiento y permitiendo crear y ejecutar con facilidad los escenarios de prueba.

Según [27], nos dice que se implementa en aplicaciones de proyectos grandes, su complejidad representa que en varios programadores trabajen simultáneamente. Su estructura es:

**a. Modelo (Model)**

Abarca la lógica de la aplicación. Si hablamos de aplicaciones web, nos referimos a servlets o servicio web. Tiene comunicación con el Presentador de tal manera que envía la información y recibe de estas consultas.

**b. Vista (View)**

Esta incluido todos los widgets que el usuario necesita para que interactúe con la aplicación. Al realizar una acción el usuario cualquiera a través de la interfaz del usuario, se encarga dicho componente de informar al Presentador lo que está ocurriendo en dicho evento y a la vez indicarle a la Vista la actualización.

**c. Presentador (Presenter)**

Este es fundamental esta pieza, es un puente que conecta al Modelo y la Vista. Responde al evento que hace el usuario, se realiza la comunicación al Modelo y depende de lo que se responde le envía a la Vista actualizarse.

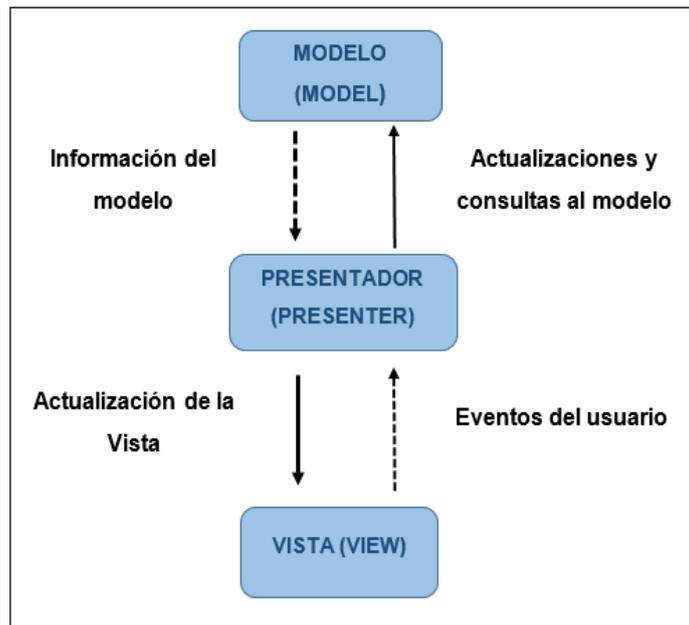


Figura 2: Esquema del patrón de diseño Modelo, Vista y Presentador (MVP). Fuente: [27]

## 2) Patrón de diseño Modelo, Vista y Controlador (MVC)

El autor [26], dice que el patrón divide los datos de la lógica de negocio con la interfaz de usuario y del módulo que hace la gestión de comunicar y eventos, reutiliza el código y separa los conceptos, la cual es característica que ayuda a realizar una búsqueda más fácil al desarrollar aplicaciones y realizar un mantenimiento posterior.

Está formado su esquema en tres componentes:

#### **a. Modelo (Model)**

El autor [26], nos dice que representa la información que opera en el sistema, la cual se encarga de gestionar el acceso de información, ya sean consultas y actualizaciones, además se implementan privilegios al acceder que fueron descritos en los requerimientos.

Según [16], dice que se encarga de avisar a las vistas la modificación en datos que ocasiona un agente externo.

#### **b. Controlador (Controller)**

El autor [26], realiza la contestación de eventos (habitualmente las acciones que hace el usuario) y también invocar las solicitudes al Modelo al efectuarse alguna petición.

Según [16] acogerá este componente los eventos de entrada, uno de los ejemplos es al dar un click, si hay una modificación en el campo de texto o más.

#### **c. Vista (View)**

El autor [26], dice que presenta al Modelo en un adecuado formato para su interacción (es la interfaz de usuario mayormente), es ahí la necesidad del Modelo para representar la salida como información.

El autor [16], nos dice que hace la recepción de datos que envía el controlador o modelo, y así posteriormente se muestre al usuario.

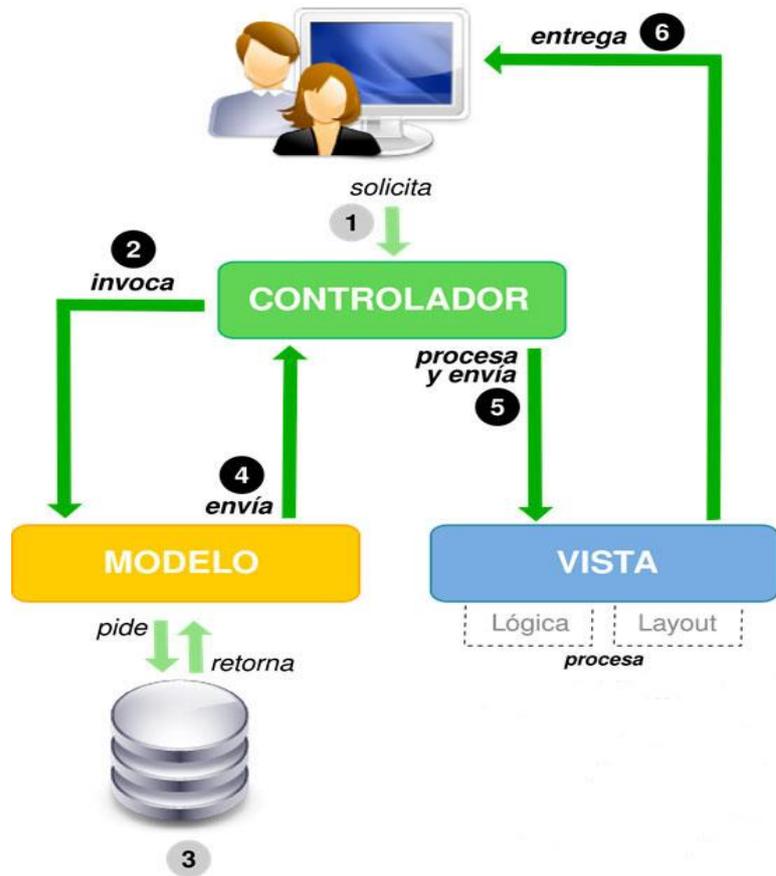


Figura 3: Esquema del patrón de diseño Modelo, Vista y Controlador (MVC). Fuente: [26]

### **3) Patrón de diseño Precontrolador, Controlador, Modelo, Vista y Postcontrolador (MVC Extendible)**

Según el autor [1], este patrón está dividido en cinco componentes que es el precontrolador, controlador, modelo, vista y postcontrolador, obteniendo como objetivo que el controlador funcione sin ningún problema.

#### **a. Precontrolador**

Procesa la solicitud del usuario, es decir acepta lo que solicita para conseguir datos URI, inventa una instancia de la clase, así misma invoca al verificador consiguiendo el tipo de método, después se manda al método de envío para la activación del controlador junto al método que se registró.

#### **b. Controlador**

Son tres pasos, comienza en el instante del método de "acción" al relacionarse con el modelo con el fin de obtener dichos datos, después el mismo método concede estos datos con un archivo de vista al método de renderización y así producir el html.

#### **c. Modelo**

Contiene todo tipo de concepto e información vinculada con el negocio, tiene comunicación con la base de datos y así una recuperación de estos mismos para efectuar las necesidades del usuario.

#### d. Vista

Una capa de presentación el cual recoge los datos que están suministrados por el controlador y produce como consecuencia html, utilizando un archivo plantilla vista; también muestra los datos conseguimos del controlador.

#### e. Postcontrolador

Su función es aceptar la salida html el cual se generó de la clase de aplicación para enviarla en el componente de respuesta e imprimirlo al usuario, primeramente, los servicios registrados son repartidos por la clase de aplicación, esta realiza el envío de vista html proveniente del componente ruta.

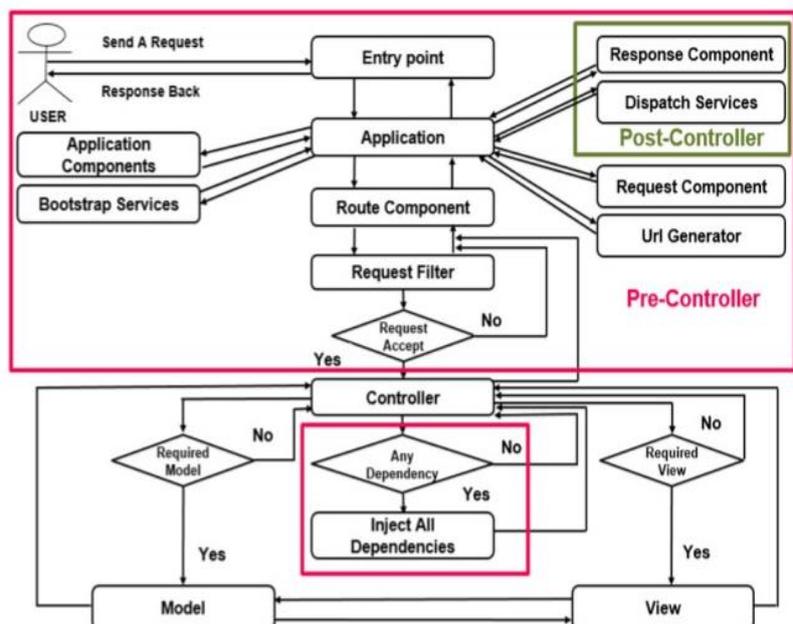


Figura 4: Esquema del patrón de diseño Precontrolador, Controlador, Modelo, Vista y Postcontrolador (MVC Extendible). Fuente: [1]

#### **4) Patrón de diseño Modelo, Vista y Vista Modelo (MVVM)**

Según el autor [4], este patrón de diseño se extiende del patrón MVC.

##### **a. Modelo (Model)**

Esta encargada de empaquetar los datos y la información, para así leerlos a través VistaModelo.

##### **b. Vista (View)**

Se comunica a menudo con el cliente y le muestra al cliente toda la información, para que los usuarios obtengan una experiencia favorable. Estas páginas pueden estar diseñadas con un simple acuerdo de vista que están basados en los requerimientos del usuario. Tiene modificaciones frecuentemente.

##### **c. Vista Modelo (View Model)**

En este componente se empaqueta la lógica de la visualización. Es encargada de la coordinación de los controles Vista y objetos que se encuentran en el Modelo. Es el principal componente para que rinda todo el sistema.

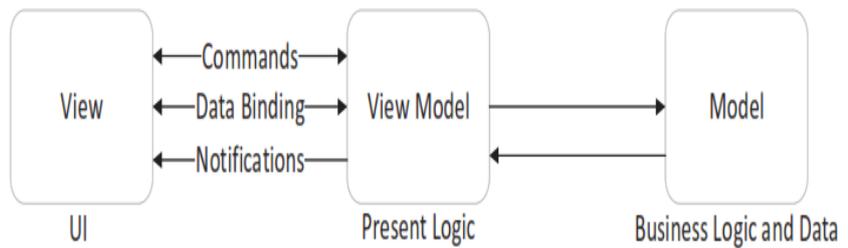


Figura 5: Esquema del patrón de diseño Modelo, Vista y Vista Modelo (MVVM). Fuente: [4]

### 5) Patrón de Diseño Presentación, Abstracción y Control (PAC)

El autor [17], dice que está basado en 3 componentes que son la presentación, abstracción y control. están separadas estas subdivisiones en aspectos de la interfaz de usuario del agente, la principal función y que se comunica que hay con los demás agentes. Para generar vistas el agente tiene la actuación de un sistema interactivo.

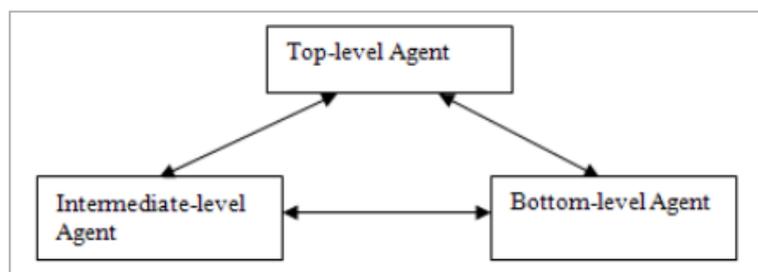


Figura 6: Esquema del patrón de diseño Presentación, Abstracción y Presentación (PAC). Fuente: [17]

Se muestra en la figura la colaboración entre los componentes, que consta de agentes separados permitiendo el desarrollo del modelo de datos e interfaz de usuario trabajando independientemente. Adentro se admiten cambios de los componentes de presentación o abstracción de un agente y además admiten múltiples tareas ya que estos agentes se distribuyen de una manera fácil a diferentes subprocesos.

## **II. MÉTODO**

### **2.1. Tipo y Diseño de Investigación**

#### **2.1.1. Tipo de Investigación: Tecnológica Aplicada**

Es típico de la Ingeniería y está vinculado a la innovación tecnológica, que busca como objetivo el conocimiento útil para resolver un problema específico que surge en las necesidades de la sociedad. [28]

Gracias a este tipo de investigación resolvemos problemas de la vida cotidiana, permitiendo ahorrar esfuerzo, tiempo y dinero para hacer un determinado trabajo.

#### **2.1.2. Diseño de la Investigación: Cuasi Experimental**

Este tipo de diseño de investigación es aquel que hace realiza la manipulación deliberadamente al menos una variable independiente para obtener su efecto o relación con las otras variables; se forman grupos de investigación antes del experimento.

Se diferencia con la investigación experimental al comparar las respuestas de los sujetos con grupos que no son iguales y así tener muchas diferencias. [28]

### **2.2. Variables, Operacionalización**

### 2.2.1. Operacionalización

Tabla 1.

*Operacionalización de variables*

Variable de estudio	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Ítem	Instrumento	Valores finales	Tipo de Variable	Escala de medición
<b>Variable Dependiente: Rendimiento en la aplicación web</b>	Medir la eficacia en que la aplicación web usa los recursos del sistema.	Tiempo de demora para el desarrollo	Desarrollo	Tiempo de Desarrollo	$TDE = \sum \text{Horas Trabajadas}$	Bitácora	Tiempo	Horas	Nivel alto: 7 horas a más. Nivel bajo: 7 horas a menos.
		Complejidad de códigos escritos		Líneas de Código Escritas	$LCE = \sum \text{Lineas de Codigo Escritas}$	Bitácora	Cantidad	Numero	Nivel alto: Promedio de 100 líneas de código a mas por archivo. Nivel bajo: Promedio de 100

							líneas de código a menos por archivo.
Consumo de memoria Ram al ejecutar diferentes transacciones	Uso de Memoria RAM	<b>UMR</b> = Cantidad de Memoria RAM Ocupada	Observación	Gigabyte	Megabyte		Nivel alto: 600 MB consumidos a mas por ejecución. Nivel bajo: 600 MB consumidos a menos por ejecución.
	Recursos						
Consumo de Uso de CPU al ejecutar diferentes transacciones	Uso de CPU	<b>UCP</b> = Cantidad de CPU Ocupada	Observación	Cantidad	Porcentaje		Nivel alto: 4% consumidos a mas por ejecución. Nivel bajo: 4%

						consumidos a menos.
Tiempo de Respuesta al ejecutar diferentes transacciones	<b>TDR =</b>					Nivel alto: 200 ms a mas por ejecución. Nivel bajo: 200 ms a menos.
	Tiempo de Respuesta	Tiempo transcurrido al realizar una peticion	Observaci ón	Tiempo	Milisegu ndos	

Fuente: Elaboración propia

### 2.2.2. Variables

#### a) Variable Independiente

Los Patrones de Diseño

#### b) Variable Dependiente

Rendimiento en la Aplicación Web

### 2.3. Población de estudio, muestra, muestreo y criterios de selección

#### 2.3.1. Población

Tabla 2.

*Patrones de diseño de software*

<b>Nº</b>	<b>Patrones de diseño</b>
1	Patrones de diseño orientado a objetos
2	Modelo, Vista y Presentador (MVP)
3	Modelo, Vista y Controlador (MVC)
4	Precontrolador, Controlador, Modelo, Vista y Postcontrolador (MVC Extendible)
5	Modelo, Vista y Vista Modelo (MVVM)
6	Presentación, Abstracción y Control (PAC)
7	Modelo, Vista y Controlador Variádico (MVC Variádico)

Fuente: Elaboración propia

### 2.3.2. Muestra

Se ha considerado trabajar con estos 2 patrones de diseño arquitectura de software.

Tabla 3.

*Patrones de diseño de software seleccionados*

<b>Nº</b>	<b>Patrones de diseño</b>
1	Modelo, Vista y Presentador (MVP)
2	Modelo, Vista y Controlador (MVC)

Fuente: Elaboración propia

### 2.3.3. Muestreo

Se seleccionó estos patrones ya que son patrones de diseño de arquitectura permitiendo desarrollar un software mediante estos patrones.

Tabla 4.

*Patrones de diseño de software seleccionados*

<b>Nº</b>	<b>Patrones de diseño</b>
1	Modelo, Vista y Presentador (MVP)
2	Modelo, Vista y Controlador (MVC)
3	Precontrolador, Controlador, Modelo, Vista y Postcontrolador (MVC Extendible)
4	Modelo, Vista y Vista Modelo (MVVM)
5	Presentación, Abstracción y Control (PAC)
6	Modelo, Vista y Controlador Variádico (MVC Variádico)

Fuente: Elaboración propia

#### 2.3.4. Criterios de Selección

Se seleccionó estos patrones ya que son los más utilizados para la creación de otros patrones de diseño es decir su arquitectura se basa en estos dos patrones de diseño, además de ser implementados con otros patrones de diseño.

Tabla 5.

*Patrones de diseño de software seleccionados*

Nº	Patrones de diseño
1	Modelo, Vista y Presentador (MVP)
2	Modelo, Vista y Controlador (MVC)

Fuente: Elaboración propia

#### 2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

##### 2.4.1. Técnica

a) **Observación:** Mediante esta técnica, se pretende observar el comportamiento de los patrones de diseño mediante el análisis de su comportamiento durante la implementación.

##### 2.4.2. Instrumentos

a) **Ficha de Observación:** Este tipo de instrumento se utiliza para conocer la forma en que se implementan los patrones de diseño y sus resultados, pudiendo analizar el rendimiento de cada uno.

- b) **Bitácora:** Este instrumento se utiliza para hacer un registro primario de los métodos a analizar.

## 2.5. Procedimientos de análisis de datos

Se utilizarán estadísticas y fórmulas que presentan medir las variables dependientes e independientes.

### 2.5.1. Variable Dependiente

Esta variable es el rendimiento en la aplicación web, detallaremos sus indicadores de esta variable:

La calificación definida de cada patrón de diseño está definida en base a la comparación de las sumas de cada indicador que se obtenga, utilizando las siguientes formulas:

– **Desarrollo**

• **Tiempo de Desarrollo**

El objetivo del indicador es conocer que facilidades nos brinda cada patrón de diseño al desarrollar una aplicación web mediante el tiempo que se demora (horas).

Tabla 6.

*Descripción del Indicador Tiempo de Desarrollo*

<b>Indicador</b>	<b>¿Qué es?</b>
TDE	Tiempo de Desarrollo

Fuente: Elaboración propia

**Formula:**  $TDE = \sum \text{Horas Trabajadas}$

Donde:  $\sum \text{Horas Trabajadas}$  es la sumatoria de las horas trabajadas para cada patrón de diseño.

- **Líneas de Código Escritas**

El objetivo del indicador es conocer que facilidades nos brinda cada patrón de diseño al desarrollar una aplicación web mediante la complejidad de códigos escritos al implementarlos.

Tabla 7.

*Descripción del Indicador Líneas de Código Escritas*

<b>Indicador</b>	<b>¿Qué es?</b>
LCE	Líneas de Código Escritas

Fuente: Elaboración propia

**Formula:**  $LCE = \sum \text{Lineas de Codigo Escritas}$

Donde:  $\sum \text{Lineas de Codigo Escritas}$  es la sumatoria de las líneas de código escritas para cada patrón de diseño.

- **Recursos**

- **Uso de Memoria de RAM**

El objetivo del indicador es determinar las características mínimas del equipo para iniciar el trabajo con cada patrón de diseño y de esta manera no tener inconvenientes mediante el uso de memoria RAM.

Tabla 8.

*Descripción del Indicador Uso de Memoria RAM*

<b>Indicador</b>	<b>¿Qué es?</b>
UMR	Uso de Memoria RAM

Fuente: Elaboración propia

**Formula:**  $UMR = \text{Cantidad de Memoria RAM Ocupada}$

Donde: Cantidad de Memoria RAM Ocupada es el espacio de memoria RAM que ocupa cada patrón de diseño.

- **Uso de CPU**

El objetivo del indicador es determinar las características mínimas del equipo para iniciar el trabajo con cada patrón de diseño y de esta manera no tener inconvenientes mediante el uso de CPU.

Tabla 9.

*Descripción del Indicador Uso de CPU*

<b>Indicador</b>	<b>¿Qué es?</b>
UCPU	Uso de CPU

Fuente: Elaboración propia

**Formula:**  $UCPU = \text{Cantidad de CPU Ocupada}$

Donde: Cantidad de CPU Ocupada es el espacio de memoria de CPU que ocupa cada patrón de diseño.

- **Tiempo de Respuesta**

El objetivo del indicador es determinar las características mínimas del equipo para iniciar el trabajo con cada patrón de diseño y de esta manera no tener inconvenientes mediante el tiempo de respuesta.

Tabla 10.

*Descripción del Indicador Tiempo de Respuesta*

<b>Indicador</b>	<b>¿Qué es?</b>
TDR	Tiempo de Respuesta

Fuente: Elaboración propia

**Formula:**  $TDR = \text{Tiempo de Respuesta}$

Donde: Tiempo de Respuesta es el tiempo transcurrido al realizar una petición para cada patrón de diseño.

## **2.6. Criterios éticos**

Los criterios éticos que se tomarán en cuenta y las acciones se realizarán para garantizarlos son los siguientes:

### **2.6.1. El consentimiento informado**

Uno de los principios éticos más importantes es el consentimiento informado ya que representa la garantía del informante que expreso voluntariamente su intención de participar en la investigación, habiendo entendido previamente la información que se le ha otorgado.

### **2.6.2. La confidencialidad**

Este definido como la garantía de que la información proporcionada está protegida y no se divulga sin el consentimiento del sujeto. Esta garantía se lleva a cabo mediante reglas que limitan el acceso a esta información.

### **2.6.3. Manejo de Riesgos**

El manejo de riesgos engloba dos aspectos principales que son: Primero, el investigador tiene que cumplir cada una de las responsabilidades y obligaciones obtenidas con los informantes, y segundo se refiere al manejo posterior de los datos que han sido proporcionado.

### **2.6.4. Criterios de rigor científico**

Los criterios de rigor científico que se tomarán en cuenta y las acciones o estrategias se realizarán para garantizarlos serán los siguientes:

– **Consistencia**

El presente trabajo es de carácter científico y formal. El análisis que se aplica a los datos se ha hecho con suma profesionalidad utilizando habilidades, técnicas y conocimientos de ingeniería e investigación para conservar la consistencia y eficacia de los datos.

– **Validez**

El resultado de esta investigación será correctamente evaluado y analizado para así obtener un resultado válido que nos ayude a que la problemática planteada sea resuelta.

– **Fiabilidad**

La investigación cumple con este principio al utilizar diferentes técnicas e instrumentos de medición que son aplicados en el proceso de recolección y transformación de la información y así brindar resultados semejantes a lo planteado al principio.

– **Transferibilidad**

La investigación proporcionara información y conocimiento que puede ser transferida a investigadores que también se enfoquen en contextos similares.

– **Neutrabilidad**

La manera en cómo se desarrolla la investigación garantiza la seguridad que los resultados que se obtuvieron, no pueden ser alterados o desviados por motivaciones, intereses, y/o perspectivas del investigador.

### III. RESULTADOS

#### 3.1. Resultados

##### 3.1.1. Determinación de la variable desarrollo

En esta variable se representarán mediante tablas y figuras el tiempo de desarrollo y líneas de código escritas al implementar cada patrón de diseño en las aplicaciones web.

###### 3.1.1.1. Tiempo de Desarrollo

Este indicador nos permite calcular el tiempo de desarrollo (horas) que se demora al implementar cada patrón de diseño al desarrollar la aplicación web.

###### – Bitácora de Tiempo de Desarrollo del Patrón de Diseño MVC

Se realizará una bitácora en el cual se detalla las horas trabajadas que se utilizó para implementar cada módulo que contiene la aplicación web.

Tabla 11.

*Bitácora de Tiempo de Desarrollo del Patrón de Diseño MVC*

Nº	Fecha	Hora		Total de horas	Responsable	Modulo	Actividades
		De	A				
1	08/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Interfaz para la página de bienvenida de la pollería y acceso al sistema (Intranet)	Se diseña la pantalla de bienvenida con los diferentes servicios que brinda la pollería y se muestra un acceso de usuario, para tener acceso se registra el USUARIO y CONTRASEÑA, luego la aplicación verifica la existencia del usuario para acceder al sistema.
		3 p.m	6 p.m		Gonzales		
	09/04/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
2	10/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Gestión de usuarios y validación de usuarios	El usuario tiene diferentes privilegios para cada módulo y se implementa el método de seguridad de acceso de usuario para conectarse con la base de datos y validar la información.
		3 p.m	6 p.m		Gonzales		
	11/04/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
3	12/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Interfaz de permisos de usuarios	Brindar diferentes permisos a los usuarios para que accedan al sistema.
		3 p.m	6 p.m		Gonzales		
	15/04/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
4	16/04/2019	9 a.m	1 p.m	14 horas			

		3 p.m	6 p.m		Christian Erick Gonzales	Registro y listado de los usuarios	Mediante una tabla se podrá visualizar a los usuarios registrados. Se podrá agregar, editar, eliminar a los usuarios.
	17/04/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	6 p.m				
<b>5</b>	18/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Interfaz de la sección clientes	Visualizar los clientes registrados y realizar búsquedas de los clientes.
		3 p.m	6 p.m				
	19/04/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	6 p.m				
<b>6</b>	22/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Registro de los clientes	Se podrá agregar, editar, eliminar a los clientes registrados en el sistema.
		3 p.m	6 p.m				
	23/04/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	6 p.m				
<b>7</b>	24/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Interfaz de la sección de productos	Visualizar los productos registrados y realizar búsquedas.
		3 p.m	6 p.m				
	25/04/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	6 p.m				
<b>8</b>	26/04/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Registro de los productos	Se podrá agregar, editar, eliminar a los productos registrados en el sistema.
		3 p.m	6 p.m				
	29/04/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	6 p.m				
<b>9</b>	30/04/2019	9 a.m	1 p.m	14 horas		Interfaz de la sección de categoría de productos	
		3 p.m	6 p.m				

---

	01/05/2019	9 a.m	1 p.m		Christian Erick Gonzales Gonzales		Visualizar la categoría de los productos registrados y realizar búsquedas de las categorías de los productos.
<b>10</b>	02/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales Gonzales	Registro de las categorías de productos	Se podrá agregar, editar, eliminar la categoría de los productos registrados en el sistema.
	03/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>11</b>	06/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales Gonzales	Interfaz de la sección de mesas	Visualizar las mesas registradas.
	07/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>12</b>	08/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales Gonzales	Registro de las mesas	Se podrá agregar, editar, eliminar las mesas registradas en el sistema.
	09/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				

---

<b>13</b>	10/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Interfaz de reservaciones de mesas vía web	Al acceder a la página web de la pollería, se encuentra una opción de RESERVAS al cual podrán los usuarios registrarse para reservar una mesa. Al completar los datos requeridos se mostrará un código de verificación, ese código será copiado por el usuario y lo llevará a la pollería para que sea verificado y confirmado su reserva.
		3 p.m	6 p.m		Gonzales		
	13/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>14</b>	14/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Visualización y registro de reservaciones de mesas vía web	Visualizar las reservaciones de las mesas vía web y se podrá agregar y editar las reservaciones de mesas vía web registradas en el sistema.
		3 p.m	6 p.m		Gonzales		
	15/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>15</b>	16/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Interfaz de la sección ventas directas y ventas por mesa	Visualizar los ventas directas y ventas por mesas registradas.
		3 p.m	6 p.m		Gonzales		
	17/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>16</b>	20/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales	Registro de las ventas directas	Se podrá agregar, editar y eliminar las ventas directas registradas en el sistema.
		3 p.m	6 p.m		Gonzales		
	21/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				

<b>17</b>	22/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales Gonzales	Registro de las ventas por mesa	Se podrá agregar, editar y eliminar los productos que el cliente desea adquirir y posteriormente realizar la venta por mesa.
		3 p.m	6 p.m				
	23/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>18</b>	24/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales Gonzales	Interfaz de pedidos online	En la página web de la pollería, se encuentra diferentes productos en venta, al cual podrán los usuarios seleccionar la imagen del pedido y así reservar dicho pedido. Al completar los datos requeridos se mostrará un código de verificación, el código será copiado por el usuario y lo llevará a la pollería para que sea verificado y confirmado su pedido.
		3 p.m	6 p.m				
	27/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>19</b>	28/05/2019	9 a.m	1 p.m	14 horas	Christian Erick Gonzales Gonzales	Registro de pedidos online	Se podrá seleccionar el producto que el cliente desea adquirir y posteriormente realizar el pedido vía web.
		3 p.m	6 p.m				
	29/05/2019	9 a.m	1 p.m				
		3 p.m	6 p.m				
<b>20</b>	30/05/2019	9 a.m	1 p.m	7 horas	Christian Erick Gonzales Gonzales	Interfaz de contactar a la pollería	Visualizar los comentarios o sugerencias que los clientes han experimentado en la pollería.
		3 p.m	6 p.m				
<b>21</b>	31/05/2019	9 a.m	1 p.m	7 horas			

---

3 p.m	6 p.m	Christian Erick Gonzales Gonzales	Registro de contactar a la pollería	Al acceder a la página web de la pollería, se encuentra una opción de CONTACTAR al cual podrán los cliente mediante un formulario enviar comentarios o sugerencias que han experimentado en la pollería.
-------	-------	---	--	--

---

Fuente: Elaboración propia

Para calcular el tiempo de desarrollo en el patrón de diseño MVC, se calculó las horas trabajadas que se utilizó para implementar el patrón, obteniendo el total de 320 horas.

**Horas diarias trabajadas:** 7 horas

**Total de días trabajadas:** 40 días

**Total de horas trabajadas al implementar el Patrón de Diseño MVC:**

7 horas x 40 días = 280 horas

– **Bitácora del Patrón de Diseño MVP**

Se realizará una bitácora en el cual se detalla las horas trabajadas que se utilizó para implementar cada módulo que contiene la aplicación web.

Tabla 12.

*Bitácora del Patrón de Diseño MVP*

Nº	Fecha	Hora		Total de horas	Responsable	Modulo	Actividades
		De	A				
1	03/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Interfaz para la página de bienvenida de la pollería y acceso al sistema (Intranet)	Se diseña la pantalla de bienvenida con los diferentes servicios que brinda la pollería y se muestra un acceso de usuario, para tener acceso se registra el USUARIO y CONTRASEÑA, luego la aplicación verifica la existencia del usuario para acceder al sistema.
		3 p.m	7 p.m				
	04/06/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
2	05/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Gestión de usuarios y validación de usuarios	El usuario tiene diferentes privilegios para cada módulo y se implementa el método de seguridad de acceso de usuario para conectarse con la base de datos y validar la información.
		3 p.m	7 p.m				
	06/06/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
3	07/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Interfaz de permisos de usuarios	Brindar diferentes permisos a los usuarios para que accedan al sistema.
		3 p.m	7 p.m				
	10/06/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	7 p.m				

<b>4</b>	11/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Registro y listado de los usuarios	Mediante una tabla se podrá visualizar a los usuarios registrados. Se podrá agregar, editar, eliminar a los usuarios.
		3 p.m	7 p.m				
	12/06/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>5</b>	13/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Interfaz de la sección clientes	Visualizar los clientes registrados y realizar búsquedas de los clientes.
		3 p.m	7 p.m				
	14/06/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>6</b>	17/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Registro de los clientes	Se podrá agregar, editar, eliminar a los clientes registrados en el sistema.
		3 p.m	7 p.m				
	18/06/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>7</b>	19/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Interfaz de la sección de productos	Visualizar los productos registrados y realizar búsquedas.
		3 p.m	7 p.m				
	20/06/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				

<b>8</b>	21/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Registro de los productos	Se podrá agregar, editar, eliminar a los productos registrados en el sistema.
		3 p.m	7 p.m				
	24/06/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	7 p.m				
<b>9</b>	25/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Interfaz de la sección de categoría de productos	Visualizar la categoría de los productos registrados y realizar búsquedas de las categorías de los productos.
		3 p.m	7 p.m				
	26/06/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	7 p.m				
<b>10</b>	27/06/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Registro de las categorías de productos	Se podrá agregar, editar, eliminar la categoría de los productos registrados en el sistema.
		3 p.m	7 p.m				
	28/06/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	7 p.m				
<b>11</b>	01/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales	Interfaz de la sección de mesas	Visualizar las mesas registradas.
		3 p.m	7 p.m				
	02/07/2019	9 a.m	1 p.m		Gonzales		
		3 p.m	7 p.m				
<b>12</b>	03/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Registro de las mesas	Se podrá agregar, editar, eliminar las mesas registradas en el sistema.
		3 p.m	7 p.m				
	04/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				

<b>13</b>	05/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Interfaz de reservaciones de mesas vía web	Al acceder a la página web de la pollería, se encuentra una opción de RESERVAS al cual podrán los usuarios registrarse para reservar una mesa. Al completar los datos requeridos se mostrará un código de verificación, ese código será copiado por el usuario y lo llevará a la pollería para que sea verificado y confirmado su reserva.
		3 p.m	7 p.m				
08/07/2019	9 a.m	1 p.m					
	3 p.m	7 p.m					
<b>14</b>	09/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Visualización y registro de reservaciones de mesas vía web	Visualizar las reservaciones de las mesas vía web y se podrá agregar y editar las reservaciones de mesas vía web registradas en el sistema.
	10/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>15</b>	11/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Interfaz de la sección ventas directas y ventas por mesa	Visualizar los ventas directas y ventas por mesas registradas.
	12/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>16</b>	15/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Registro de las ventas directas	Se podrá agregar, editar y eliminar las ventas directas registradas en el sistema.
	16/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				

<b>17</b>	17/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Registro de las ventas por mesa	Se podrá agregar, editar y eliminar los productos que el cliente desea adquirir y posteriormente realizar la venta por mesa.
		3 p.m	7 p.m				
	18/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>18</b>	19/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Interfaz de pedidos online	En la página web de la pollería, se encuentra diferentes productos en venta, al cual podrán los usuarios seleccionar la imagen del pedido y así reservar dicho pedido. Al completar los datos requeridos se mostrará un código de verificación, el código será copiado por el usuario y lo llevará a la pollería para que sea verificado y confirmado su pedido.
		3 p.m	7 p.m				
	22/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>19</b>	23/07/2019	9 a.m	1 p.m	16 horas	Christian Erick Gonzales Gonzales	Registro de pedidos online	Se podrá seleccionar el producto que el cliente desea adquirir y posteriormente realizar el pedido vía web.
		3 p.m	7 p.m				
	24/07/2019	9 a.m	1 p.m				
		3 p.m	7 p.m				
<b>20</b>	25/07/2019	9 a.m	1 p.m	8 horas	Christian Erick Gonzales Gonzales	Interfaz de contactar a la pollería	Visualizar los comentarios o sugerencias que los clientes han experimentado en la pollería.
		3 p.m	7 p.m				

<b>21</b>	26/07/2019	9 a.m	1 p.m	8 horas	Christian Erick Gonzales Gonzales	Registro de contactar a la pollería	Al acceder a la página web de la pollería, se encuentra una opción de CONTACTAR al cual podrán los cliente mediante un formulario enviar comentarios o sugerencias que han experimentado en la pollería.
-----------	------------	-------	-------	---------	---	--	--

Fuente: Elaboración propia

Para calcular el tiempo de desarrollo en el patrón de diseño MVP, se calculó las horas trabajadas que se utilizó para implementar el patrón, obteniendo el total de 154 horas.

**Horas diarias trabajadas:** 8 horas

**Total de días trabajadas:** 40 días

**Total de horas trabajadas al implementar el Patrón de Diseño MVP:**

8 horas x 40 días = 320 horas

– **Interpretación del indicador del Tiempo de Desarrollo**

Tabla 13:

*Resultados obtenidos respecto al Tiempo de Desarrollo*

<b>Patrones de diseño</b>	<b>Tiempo (H)</b>	<b>Numero de Historias de Usuario</b>
Modelo, Vista y Controlador (MVC)	280 horas	21
Modelo, Vista y Presentador (MVP)	320 horas	21

Fuente: Elaboración propia

### **3.1.1.2. Líneas de código escritas**

Este indicador nos permitirá realizar la evaluación de la cantidad de líneas de código que tiene cada proceso con el patrón de diseño MVC y MVP, que tiene como objetivo establecer que patrón de diseño es el que menos líneas de código genera al implementarse.

#### **– Conteo de líneas del Patrón de Diseño MVC**

Se realizará una tabla en el cual se detalla el número de líneas escritas por cada módulo que se utilizó para implementar la aplicación web.

Tabla 14.

*Bitácora de Líneas de Código Escritas del Patrón de Diseño MVC*

	<b>Ruta de Ubicación</b>	<b>Archivo</b>	<b>Líneas de Código Escritos</b>
	sistema_polleria/ <b>modelo/</b>	modelo_categorias.php	76 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_clientes.php	51 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_contacto.php	40 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_mesas.php	77 LDCE
<b>Modelo</b>	sistema_polleria/ <b>modelo/</b>	modelo_pedidosonline.php	40 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_productos.php	75 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_reservas.php	51 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_ventas.php	137 LDCE
	sistema_polleria/ <b>modelo/</b>	modelo_ventaspormesa.php	25 LDCE

---

	controlador_comboscategoria.php	
sistema_polleria/ <b>controlador</b> /categorias/	controlador_editar_categorias.php	41 LDCE
	controlador_ListarBuscar_categorias.php	
	controlador_registrar_categoria.php	
	controlador_editar_clientes.php	
sistema_polleria/ <b>controlador</b> /clientes/	controlador_ListarBuscar_clientes.php	39 LDCE
	controlador_registrar_clientes.php	
<b>Controlador</b>		
sistema_polleria/ <b>controlador</b> /contacto/	controlador_ListarBuscar_contacto.php	27 LDCE
	controlador_registrar_contacto.php	
	controlador_combolistar_mesas.php	
	controlador_editar_mesas.php	
sistema_polleria/ <b>controlador</b> /mesas/	controlador_listar_mesas.php	55 LDCE
	controlador_ListarBuscar_mesas.php	
	controlador_registrar_mesas.php	
sistema_polleria/ <b>controlador</b> /pedidos/	controlador_ListarBuscar_pedidosonline.php	40 LDCE

---

---

	controlador_registrar_pedido_web.php	
	controlador_edit_productos.php	
sistema_polleria/controlador/productos/	controlador_eliminar_productos.php	56 LDCE
	controlador_ListarBuscar_productos.php	
	controlador_registrar_producto.php	
sistema_polleria/controlador/reservas/	controlador_ListarBuscar_reservas.php	43 LDCE
	controlador_registrar_reserva_web.php	
	controlador_buscar_cliente.php	
	controlador_buscar_producto.php	
sistema_polleria/controlador/pedidos/	controlador_buscar_producto_mesa.php	207 LDCE
	controlador_listar.php	
	controlador_listar_tmp.php	
	controlador_listar_tmp_mesa.php	

---

---

	controlador_ListarBuscar_ventas.php	
	controlador_ListarBuscar_ventaspor mesa.php	
	controlador_registrar_producto.php	
	controlador_registrar_producto_mesa.php	
	controlador_registrar_ventas.php	
	controlador_registrar_ventasmesa.php	
	controlador_remove_producto.php	
	vista_categorias_listar.php	167 LDCE
	vista_categorias_registrar.php	
<b>Vista</b>	vista_clientes_listar.php	179 LDCE
	vista_clientes_registrar.php	
	vista_contacto_listar.php	123 LDCE

---

---

	vista_contacto_registrar.php	
sistema_polleria/ <b>vistas</b> /	contactar.php	150 LDCE
	vista_mesas_listar.php	
sistema_polleria/ <b>vistas</b> /Mesas/		200 LDCE
	vista_mesas_registrar.php	
sistema_polleria/ <b>vistas</b> /Pedidos/	vista_listar_pedidosonline.php	43 LDCE
sistema_polleria/ <b>vistas</b> /	pedido.php	173 LDCE
	vista_productos_listar.php	
sistema_polleria/ <b>vistas</b> /Productos/		232 LDCE
	vista_productos_registrar.php	
sistema_polleria/ <b>vistas</b> /Reservas/	vista_reservas_listar.php	106 LDCE
sistema_polleria/ <b>vistas</b> /	reservas.php	166 LDCE
	vista_listar_ventas.php	
sistema_polleria/ <b>vistas</b> /Pedidos/		273 LDCE
	vista_ventas_registrar.php	

---

---

sistema_polleria/ <b>vistas</b> /Pedidos/	vista_listar_ventasmesa.php	206 LDCE
	vista_ventas_registrarmesa.php	

---

Fuente: Elaboración propia

Para calcular las líneas de código escritas en el patrón de diseño MVC, se calculó contando las líneas de código que están implementadas en cada módulo que tiene la aplicación web, obteniendo el total de 3 098 LDCE.

**Total de archivos codificados:** 63 archivos

**Total de líneas escritas al implementar el Patrón de Diseño MVC:** 3 098 LDCE

– **Conteo de líneas del Patrón de Diseño MVP**

Se realizará una tabla en el cual se detalla el número de líneas escritas por cada módulo que se utilizó para implementar la aplicación web.

Tabla 15:

*Bitácora de Líneas de Código Escritas del Patrón de Diseño MVP*

	<b>Ruta de Ubicación</b>	<b>Archivo</b>	<b>Líneas de Código Escritos</b>
<b>Modelo</b>	POLLERIAMVP/admin/ <b>model</b> /catalog/	categoria.php	73 LDCE
	POLLERIAMVP/admin/ <b>model</b> /catalog/	contact.php	52 LDCE
	POLLERIAMVP/admin/ <b>model</b> /catalog/	mesas.php	83 LDCE
	POLLERIAMVP/admin/ <b>model</b> /catalog/	product.php	84 LDCE
	POLLERIAMVP/admin/ <b>model</b> /catalog/	reservas.php	72 LDCE
	POLLERIAMVP/admin/ <b>model</b> /customer/	customer.php	73 LDCE
	POLLERIAMVP/admin/ <b>model</b> /orden/	pedidosonline.php	71 LDCE
	POLLERIAMVP/admin/ <b>model</b> /orden/	ventas.php	134 LDCE

---

	POLLERIAMVP/admin/ <b>model</b> /orden/	ventasmesa.php	128 LDCE
	POLLERIAMVP/catalog/ <b>model</b> /reservation/	return.php	49 LDCE
	POLLERIAMVP/catalog/ <b>model</b> /catalog /	product.php	125 LDCE
	POLLERIAMVP/catalog/ <b>model</b> /catalog/	category.php	39 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /catalog/	categoria.php	310 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /catalog/	contact.php	130 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /catalog/	mesas.php	358 LDCE
<b>Presentador</b>	POLLERIAMVP/admin/ <b>presenter</b> /catalog/	product.php	441 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /catalog/	reservas.php	413 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /customer/	customer.php	464 LDCE

---

---

	POLLERIAMVP/admin/ <b>presenter</b> /orden/	pedidosonline.php	411 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /orden/	ventas.php	389 LDCE
	POLLERIAMVP/admin/ <b>presenter</b> /orden/	ventasmesa.php	326 LDCE
	POLLERIAMVP/catalog/ <b>presenter</b> /information/	contact.php	130 LDCE
	POLLERIAMVP/catalog/ <b>presenter</b> /product/	category.php	256 LDCE
	POLLERIAMVP/catalog/ <b>presenter</b> /product/	product.php	244 LDCE
	POLLERIAMVP/catalog/ <b>presenter</b> /reservation/	return.php	157 LDCE
	POLLERIAMVP/admin/ <b>view</b> /template/catalog/	categoría_form.tpl	42 LDCE
<b>Vista</b>	POLLERIAMVP/admin/ <b>view</b> /template/catalog/	categoría_list.tpl	76 LDCE
	POLLERIAMVP/admin/ <b>view</b> /template/catalog/	contact_list.tpl	65 LDCE

---

---

POLLERIAMVP/admin/ <b>view</b> /template/catalog/	mesa_form.tpl	74 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/catalog/	mesa_list.tpl	90 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/catalog/	product_form.tpl	185 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/catalog/	product_list.tpl	148 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/catalog/	reservas_form.tpl	118 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/catalog/	reservas_list.tpl	79 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/customer/	customer_form.tpl	78 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/customer/	customer_list.tpl	164 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/orden/	pedidosonline_form.tp l	118 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/orden/	pedidosonline_list.tpl	79 LDCE

---

---

POLLERIAMVP/admin/ <b>view</b> /template/orden/	ventas_form.tpl	350 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/orden/	ventas_list.tpl	87 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/orden/	ventasmesa_form.tpl	346 LDCE
POLLERIAMVP/admin/ <b>view</b> /template/orden/	ventasmesa_list.tpl	64 LDCE
POLLERIAMVP/catalog/ <b>view</b> /theme/template/information	contact.tpl	81 LDCE
POLLERIAMVP/catalog/ <b>view</b> /theme/template/product	category.tpl	45 LDCE
POLLERIAMVP/catalog/ <b>view</b> /theme/template/product	product.tpl	170 LDCE
POLLERIAMVP/catalog/ <b>view</b> /theme/template/reservation	return_form.tpl	86 LDCE

---

Fuente: Elaboración propia

Para calcular las líneas de código escritas en el patrón de diseño MVP, se calculó contando las líneas de código que están implementadas en cada módulo que tiene la aplicación web, obteniendo el total de 7 587 LDCE.

**Total de archivos codificados:** 46 archivos

**Total de líneas escritas al implementar el Patrón de Diseño MVP:** 7 384 LDCE

– **Interpretación del indicador de Líneas de Código Escritas**

Tabla 16:

*Resultados obtenidos respecto al indicador Líneas de Código Escritas*

<b>Patrones de diseño</b>	<b>Líneas de Código Escritas (LDCE)</b>	<b>Archivos</b>
Modelo, Vista y Controlador (MVC)	3 098 LDCE	63
Modelo, Vista y Presentador (MVP)	7 587 LDCE	46

Fuente: Elaboración propia

### 3.1.2. Interpretación de los resultados de la variable desarrollo

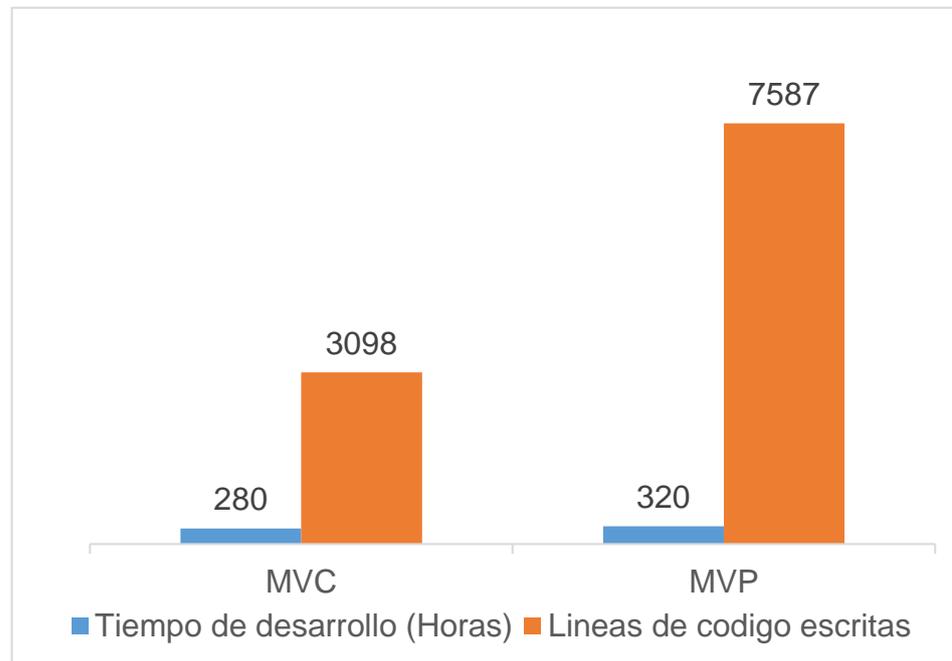


Figura 7: Interpretación de la Variable Desarrollo. Fuente: *Elaboración propia*

En la variable Desarrollo se muestra que el patrón de diseño MVC tiene ventajas con el patrón de diseño MVP en el indicador líneas de código y en el indicador de tiempo de desarrollo.

### 3.1.3. Determinación de la variable recursos

En esta variable se representarán mediante tablas y figuras el uso de memoria RAM, uso de CPU y tiempo de respuesta al implementar cada patrón de diseño en las aplicaciones web.

– **Escenario 1 para realizar las pruebas**

Para realizar las pruebas y recopilación de información, para este indicador se utilizó una computadora con las siguientes características.

Tabla 17:

*Escenario para realizar las pruebas de la Variable Recursos*

<b>Computadora</b>	
<b>Hardware</b>	
<i>Procesador</i>	Intel Pentium D 820
<i>Memoria RAM</i>	1 GB (1 024 MB)
<i>Disco duro</i>	500 GB
<b>Software</b>	
<i>Sistema operativo</i>	Windows 7
<i>Base de datos</i>	Mysql
<i>Servidor local</i>	XAMPP

Fuente: Elaboración propia

– **Herramientas para realizar las pruebas**

Para realizar las pruebas y recopilación de información, en este indicador se hace uso de la herramienta IObit SysInfo que nos permite monitorear en tiempo real la temperatura y la tasa de uso de los recursos de CPU, GPU, disco, placa base y memoria.

### 3.1.3.1. Uso de memoria RAM

Este indicador hace referencia a la cantidad de memoria RAM ocupada por la aplicación web al realizar un proceso, el cual cada aplicación esta creada e implementada con cada patrón de diseño.

Para medir este indicador se realizó diferentes pruebas al ejecutar las dos aplicaciones web elaboradas, tiene como objetivo la comparación de las características de rendimiento, en el cual se determinará el uso de memoria RAM al momento de realizar diferentes procesos en la aplicación web.

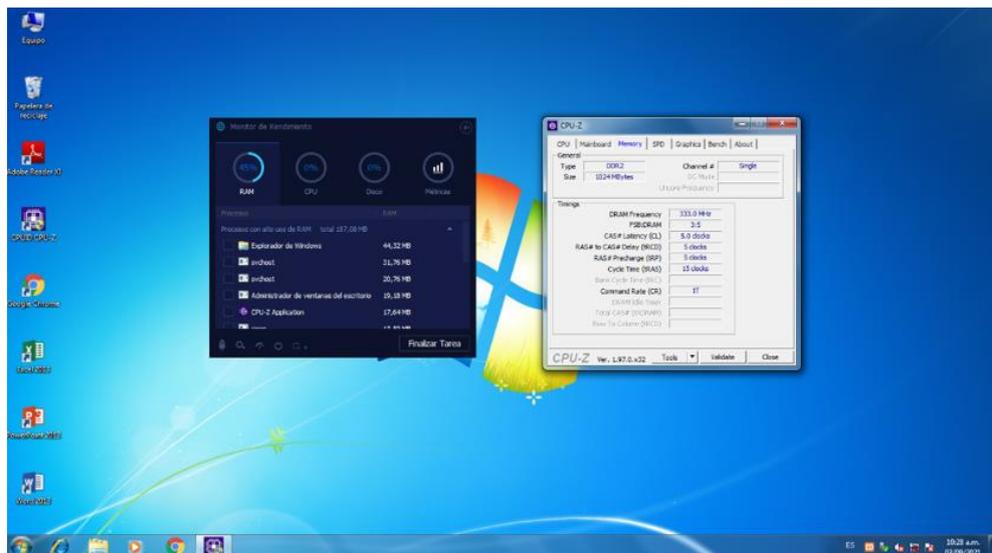


Figura 8: Memoria RAM disponible para la utilización de los patrones de diseño. Fuente: *Elaboración propia*

– Pruebas realizadas en el Patrón de diseño MVC

Se realizaron diferentes pruebas para medir el rendimiento en el uso de la memoria RAM ejecutando la aplicación web desarrollada con el patrón de diseño MVC.

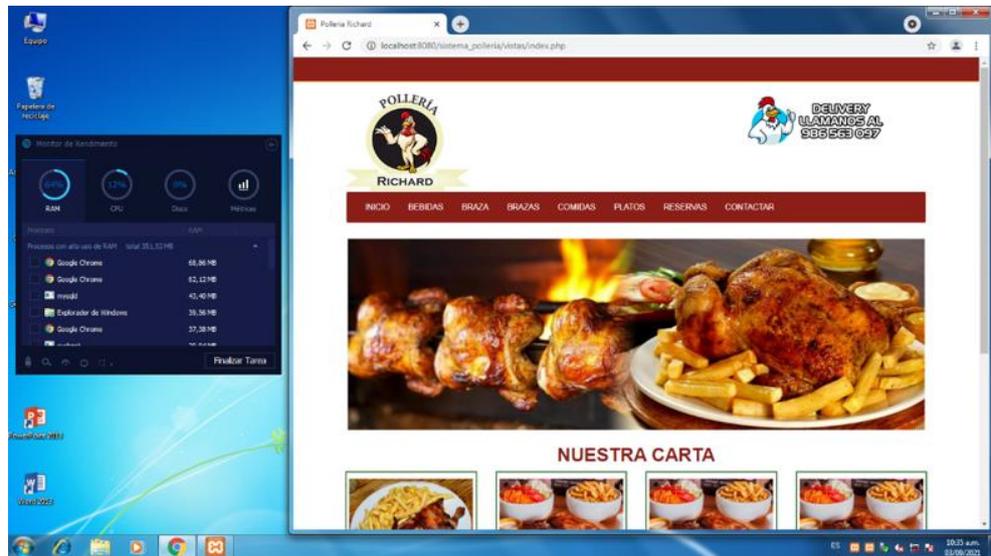


Figura 9: Utilización de Memoria RAM al ejecutar la aplicación web MVC. Fuente: *Elaboración propia*

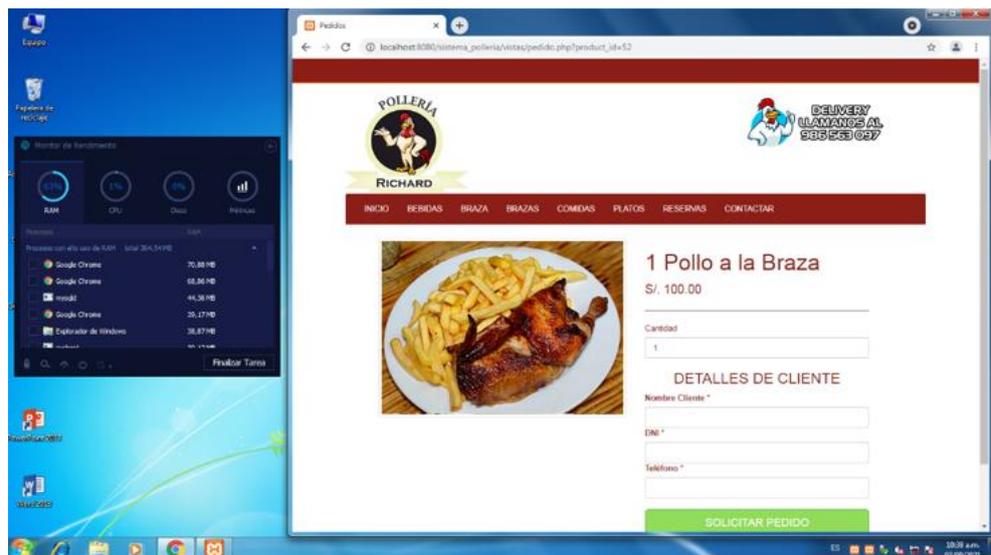


Figura 10: Utilización de Memoria RAM al ejecutar el proceso de Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

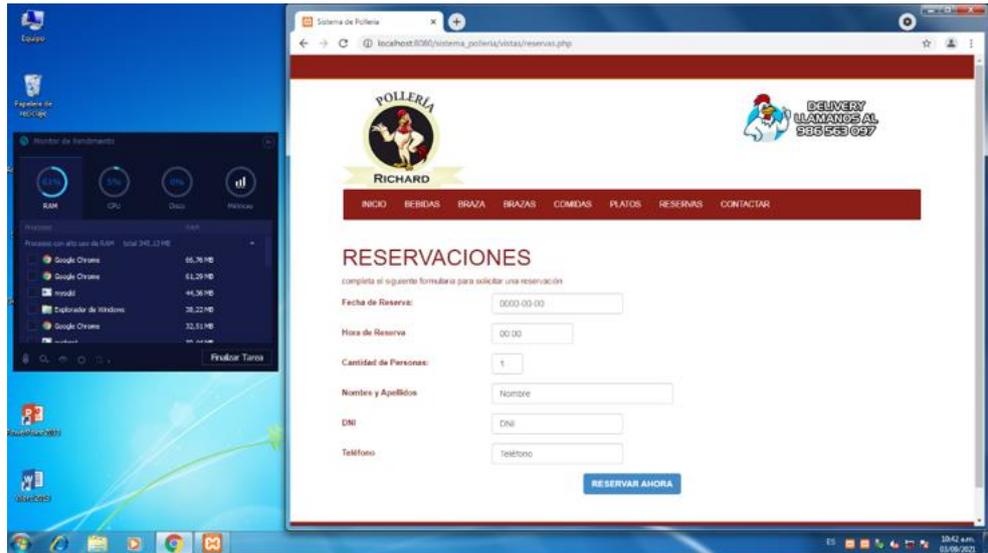


Figura 11: Utilización de Memoria RAM al ejecutar el proceso de Reservas web de la aplicación web MVC. Fuente: *Elaboración propia*

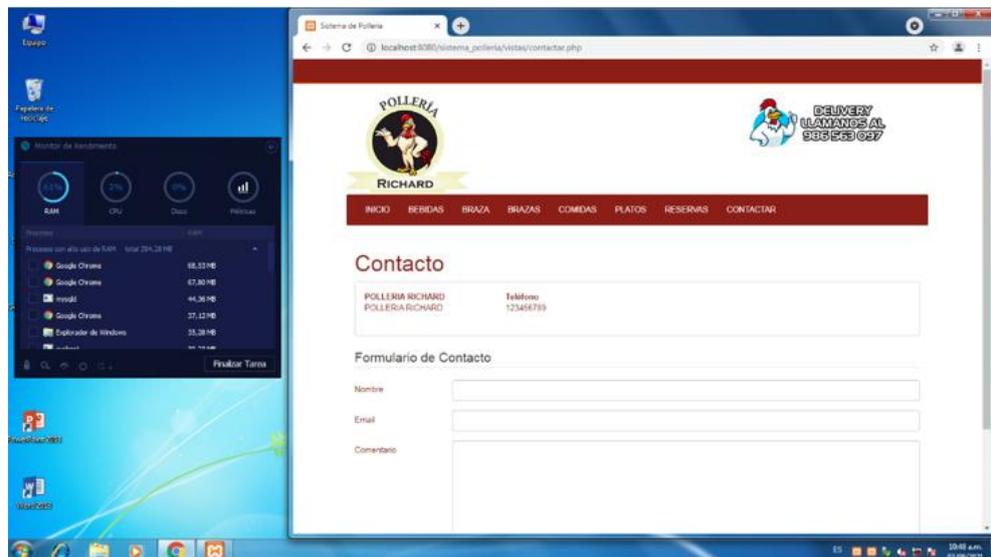


Figura 12: Utilización de Memoria RAM al ejecutar el proceso de Contactar a la pollería de la aplicación web MVC. Fuente: *Elaboración propia*

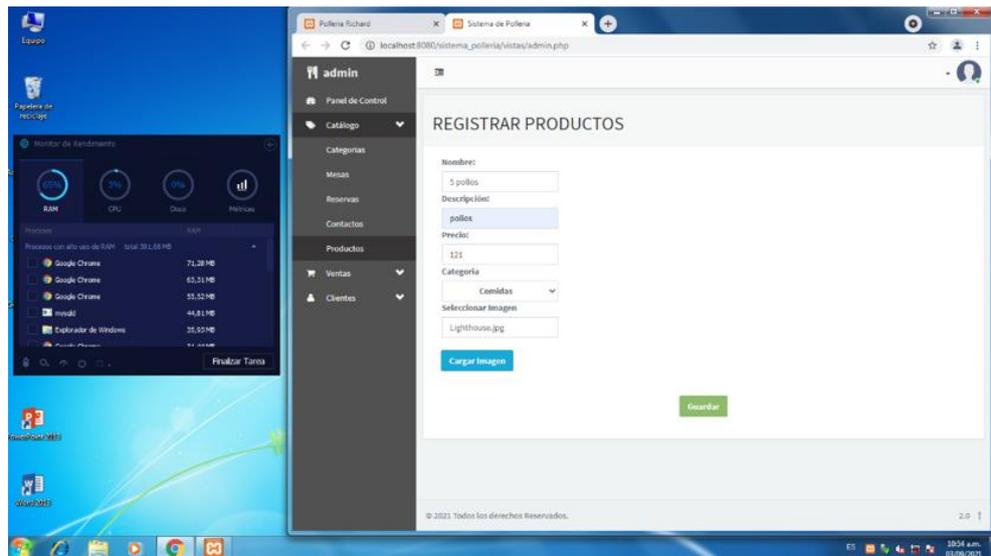


Figura 13: Utilización de Memoria RAM al ejecutar el proceso de Registrar Producto de la aplicación web MVC. Fuente: *Elaboración propia*

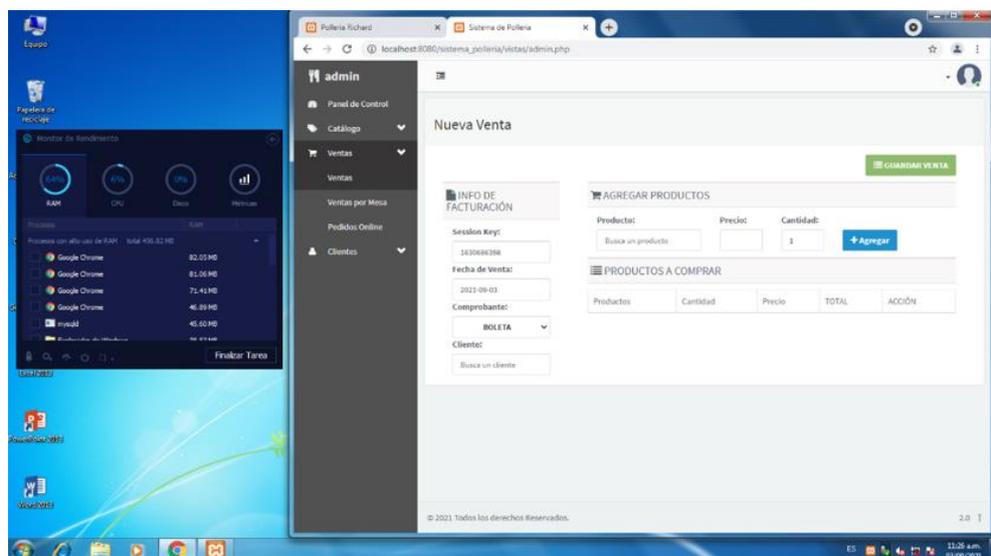


Figura 14: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVC. Fuente: *Elaboración propia*

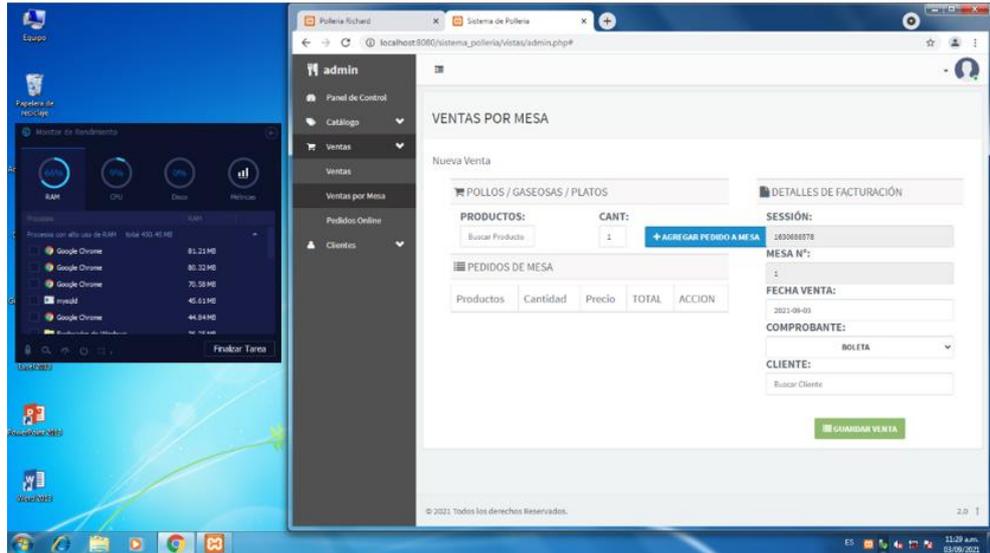


Figura 15: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVC. Fuente: *Elaboración propia*

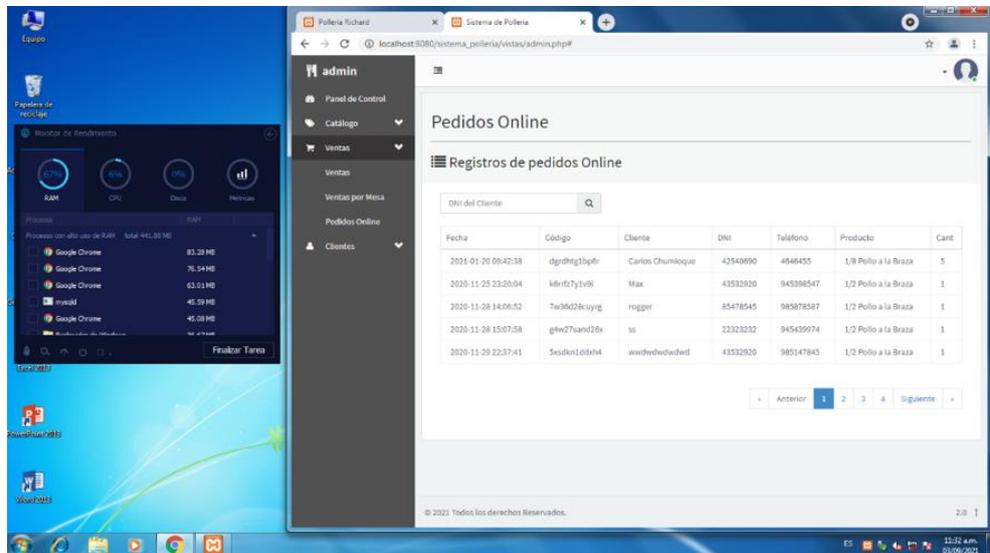


Figura 16: Utilización de Memoria RAM al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

Para calcular el uso de Memoria RAM en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso de la Memoria RAM 654 MB del total que tiene la computadora que es de 1024 MB.

**Total de proceso ejecutados:** 8 procesos

**Total de Memoria RAM utilizada al implementar el Patrón de Diseño MVC:** 654 MB

– **Pruebas realizadas en el Patrón de diseño MVP**

Se realizaron diferentes pruebas para medir el rendimiento en el uso de la memoria RAM ejecutando la aplicación web desarrollada con el patrón de diseño MVP.



Figura 17: Utilización de Memoria RAM al ejecutar la aplicación web MVP.  
Fuente: *Elaboración propia*

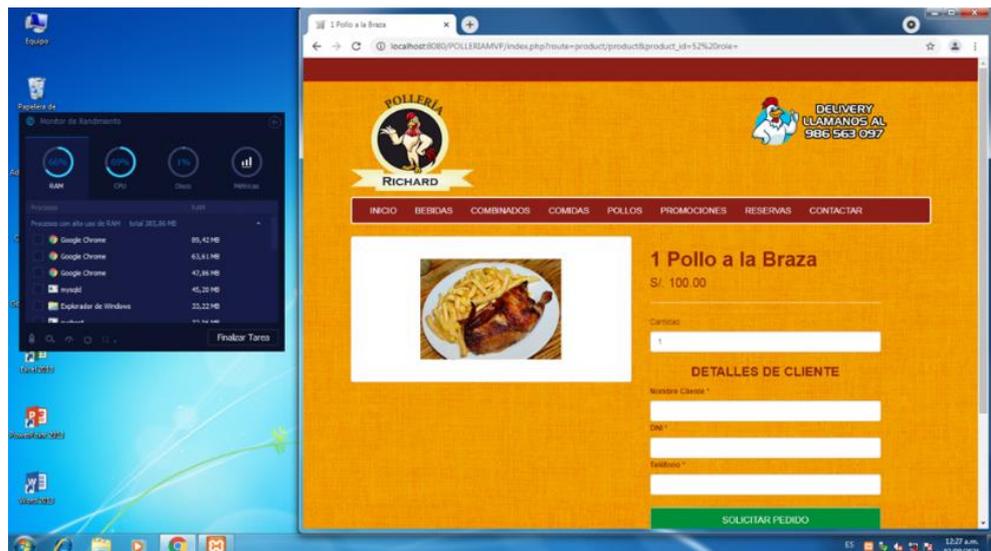


Figura 18: Utilización de Memoria RAM al ejecutar el proceso de Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

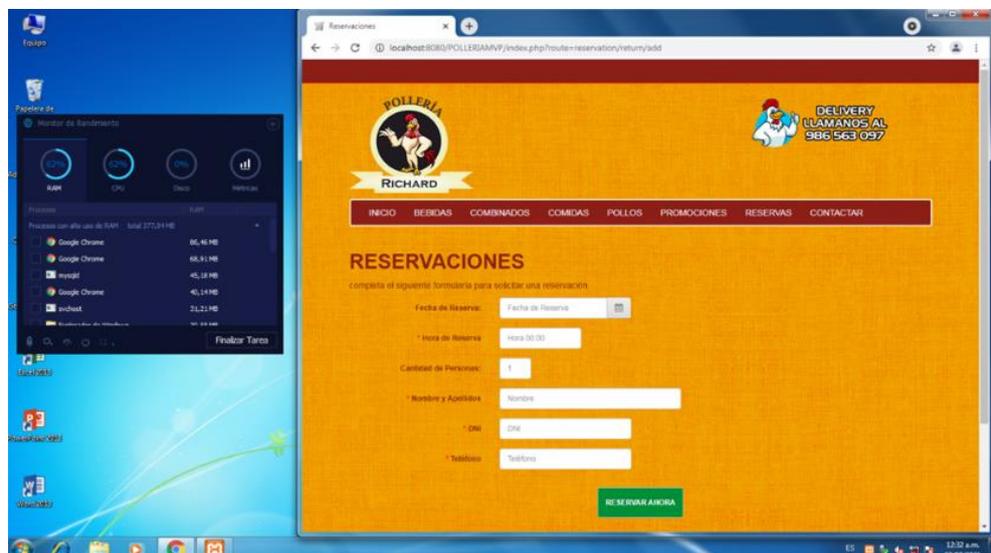


Figura 19: Utilización de Memoria RAM al ejecutar el proceso de Reservas web de la aplicación web MVP. Fuente: *Elaboración propia*

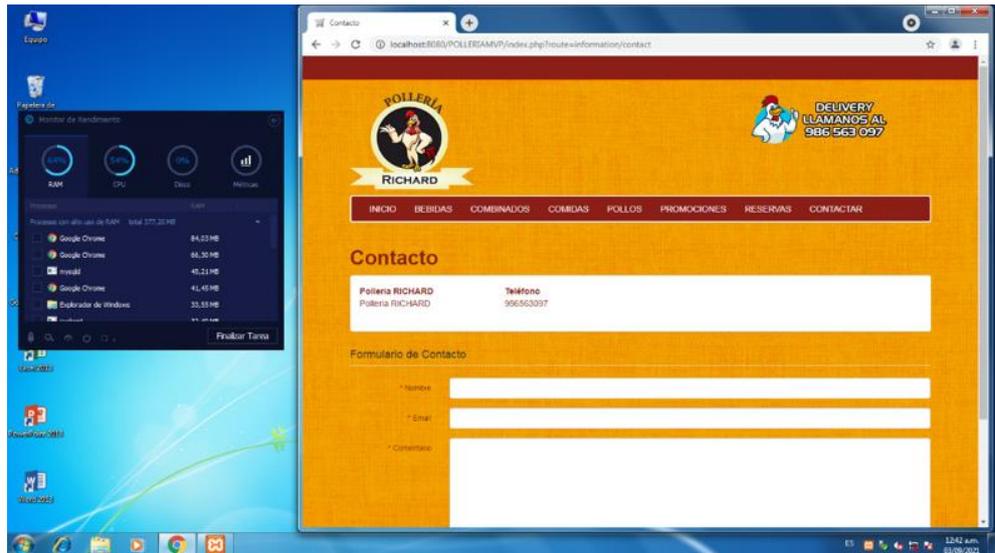


Figura 20: Utilización de Memoria RAM al ejecutar el proceso de Contactar a la pollería de la aplicación web MVP. Fuente: *Elaboración propia*

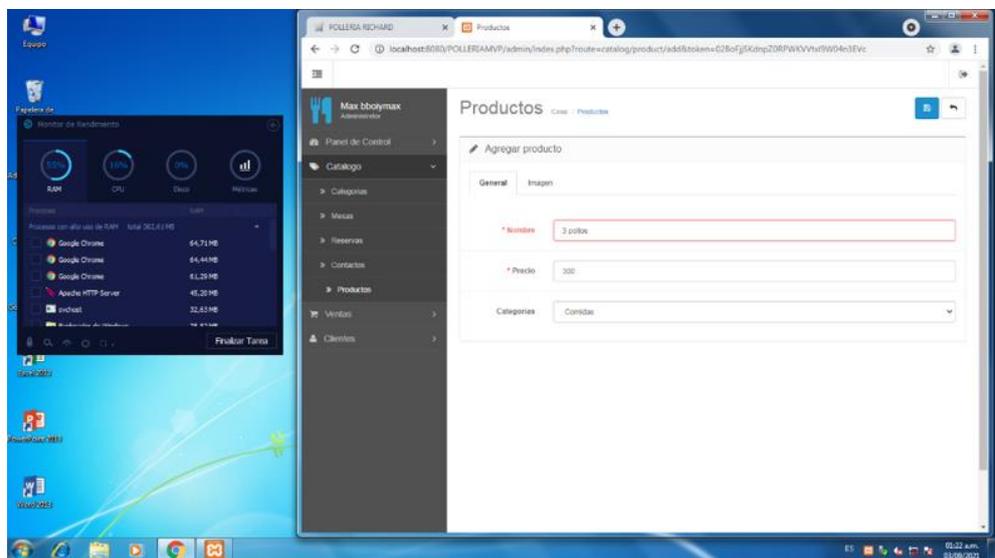


Figura 21: Utilización de Memoria RAM al ejecutar el proceso de Registrar Producto de la aplicación web MVP. Fuente: *Elaboración propia*

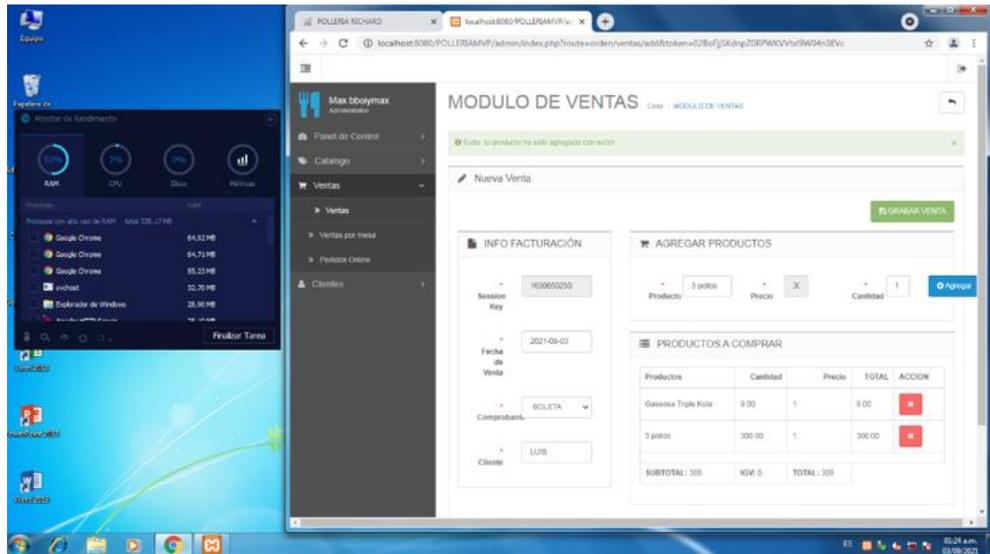


Figura 22: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVP. Fuente: *Elaboración propia*

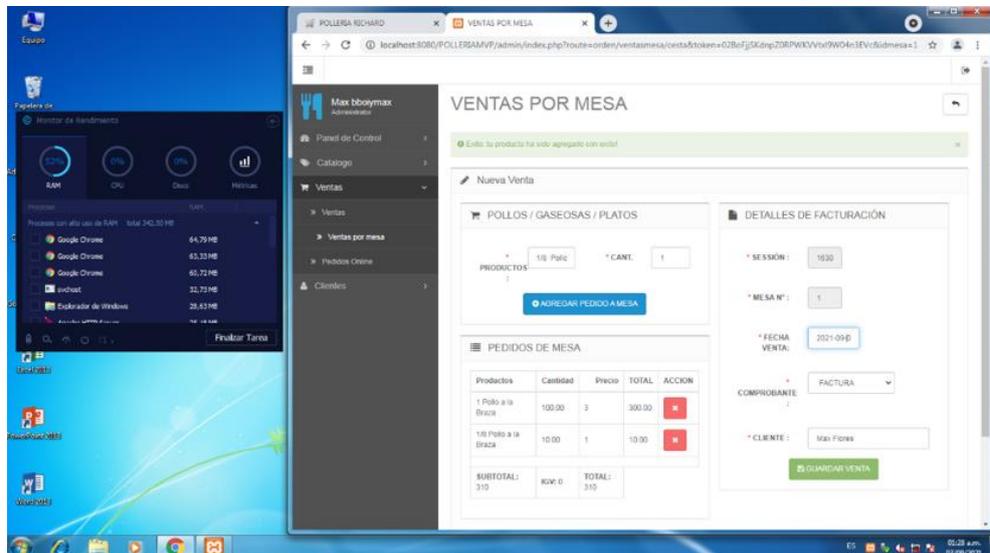


Figura 23: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVP. Fuente: *Elaboración propia*

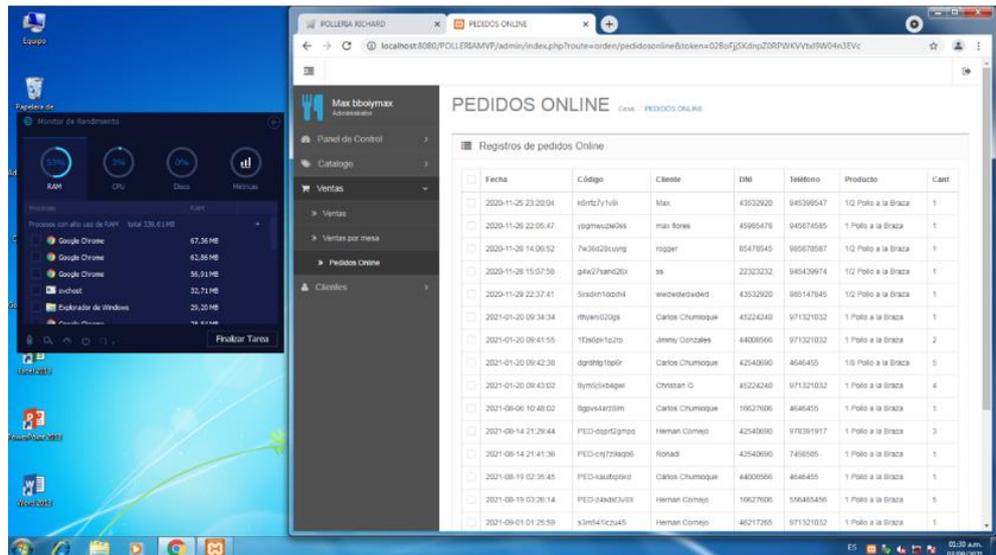


Figura 24: Utilización de Memoria RAM al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

Para calcular el uso de Memoria RAM en el patrón de diseño MVP, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso de la Memoria RAM 598 MB del total que tiene la computadora que es de 1024 MB.

**Total de proceso ejecutados: 8 procesos**

**Total de Memoria RAM utilizada al implementar el Patrón de Diseño**

**MVP: 598 MB**

– **Interpretación del indicador de Memoria Ram**

Tabla 18.

*Resultados obtenidos respecto al indicador de Memoria Ram*

<b>Patrones de diseño</b>	<b>Memoria RAM utilizada</b>	<b>Procesos realizados</b>
Modelo, Vista y Controlador (MVC)	654 MB	8
Modelo, Vista y Presentador (MVP)	598 MB	8

Fuente: Elaboración propia

– **Escenario 2 para realizar las pruebas**

Tabla 19:

*Escenario 2 para realizar las pruebas de la Variable Recursos*

<b>Laptop</b>	
<b>Hardware</b>	
<i>Procesador</i>	Intel Core i7-7500U
<i>Memoria RAM</i>	12 GB (12 000 MB)
<i>Disco duro</i>	1 000 GB
<b>Software</b>	
<i>Sistema operativo</i>	Windows 10 Home
<i>Base de datos</i>	Mysql
<i>Servidor local</i>	XAMPP

Fuente: Elaboración propia

– Pruebas realizadas en el Patrón de diseño MVC

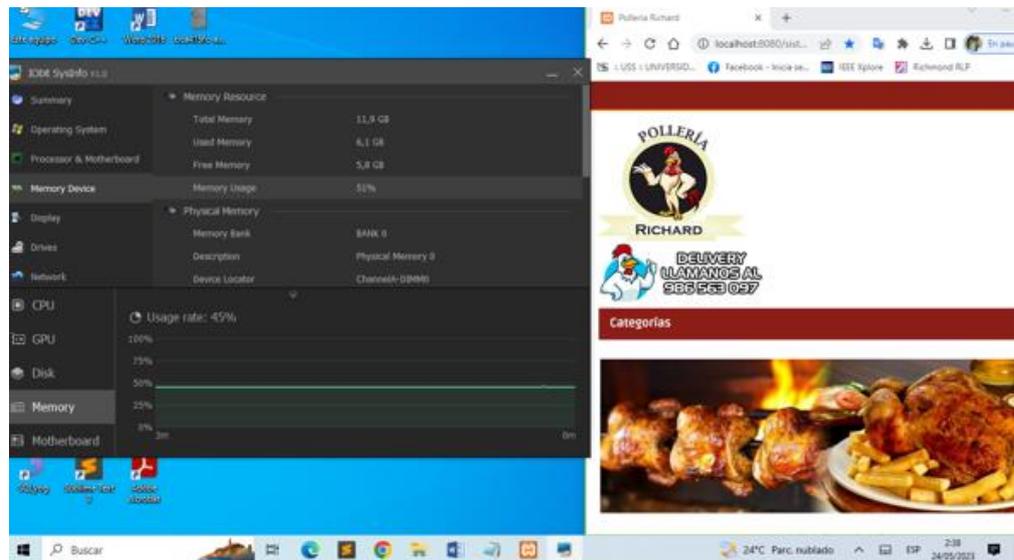


Figura 25: Memoria RAM disponible para la utilización de los patrones de diseño. Fuente: *Elaboración propia*

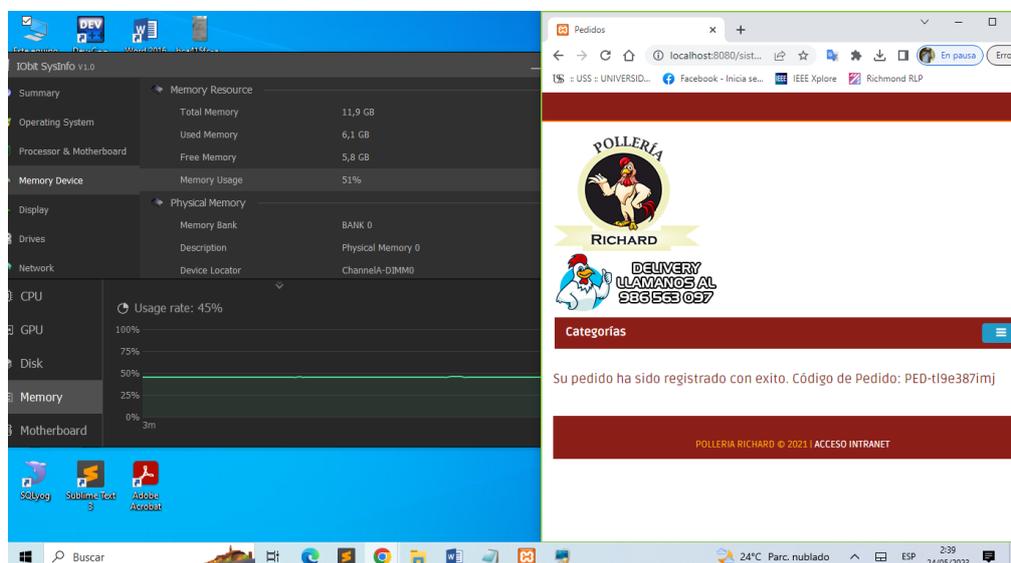


Figura 26: Utilización de Memoria RAM al ejecutar el proceso de Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

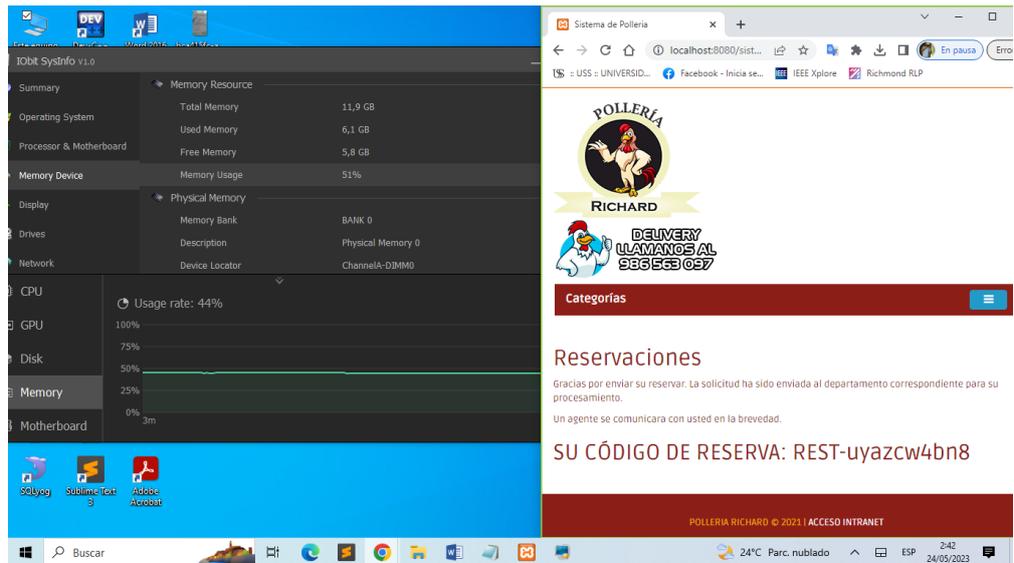


Figura 27: Utilización de Memoria RAM al ejecutar el proceso de Reservaciones web de la aplicación web MVC. Fuente: *Elaboración propia*

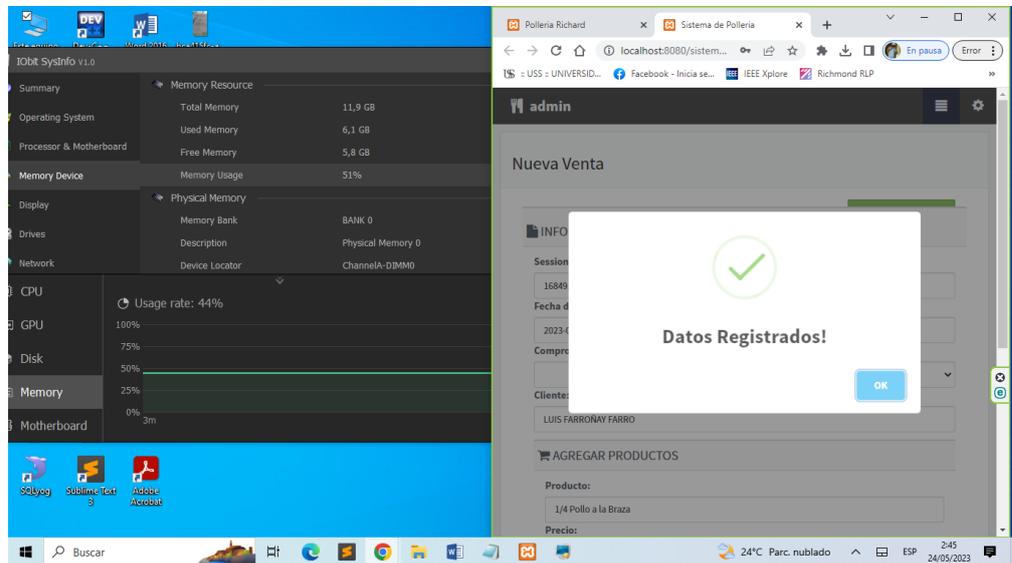


Figura 28: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVC. Fuente: *Elaboración propia*

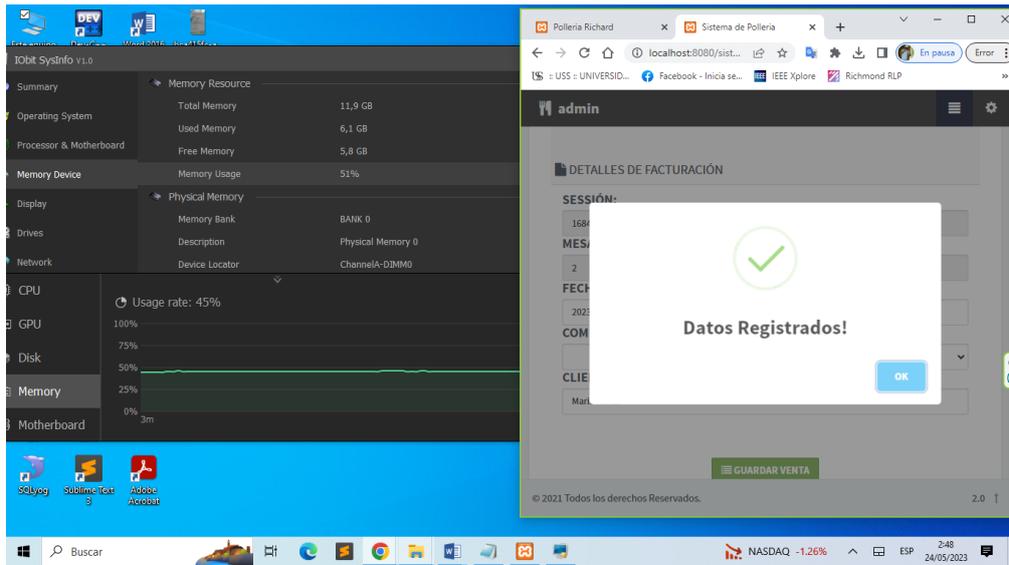


Figura 29: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVC. Fuente: *Elaboración propia*

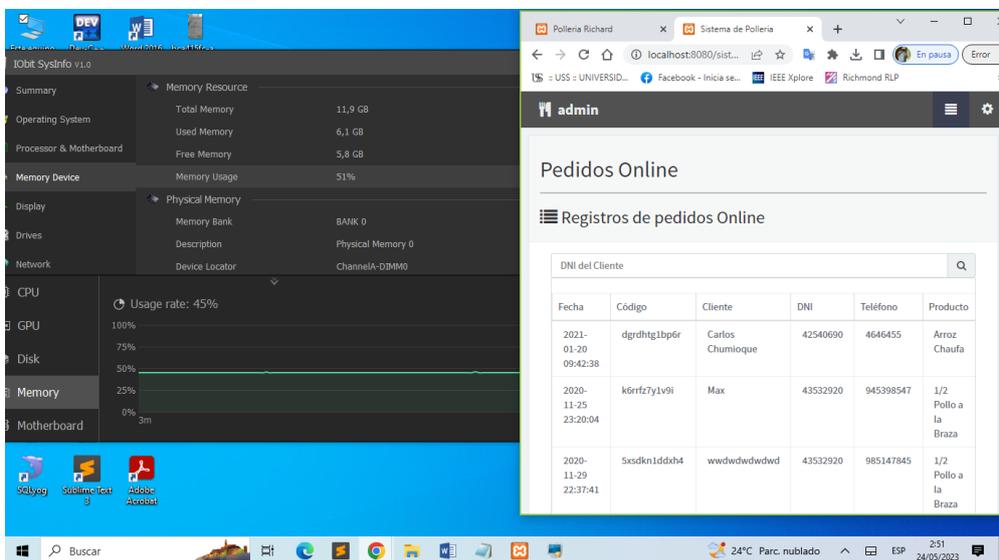
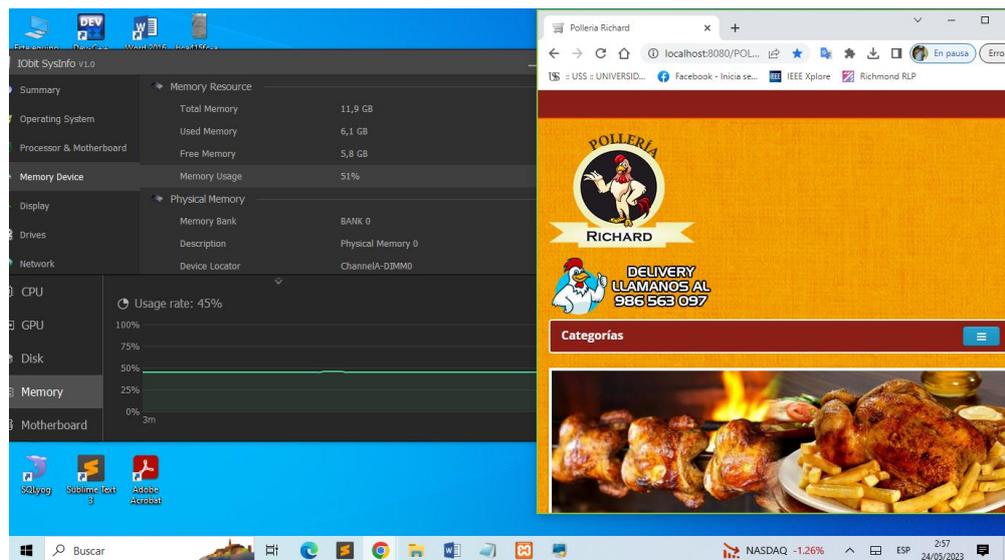


Figura 30: Utilización de Memoria RAM al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

Para calcular el uso de Memoria RAM en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso de la Memoria RAM 5 520 MB del total que tiene la laptop que es de 12 000 MB.

**Total de Memoria RAM utilizada al implementar el Patrón de Diseño MVC: 5 520 MB**

– **Pruebas realizadas en el Patrón de diseño MVP**



*Figura 31: Utilización de Memoria RAM al ejecutar la aplicación web MVP.  
Fuente: Elaboración propia*

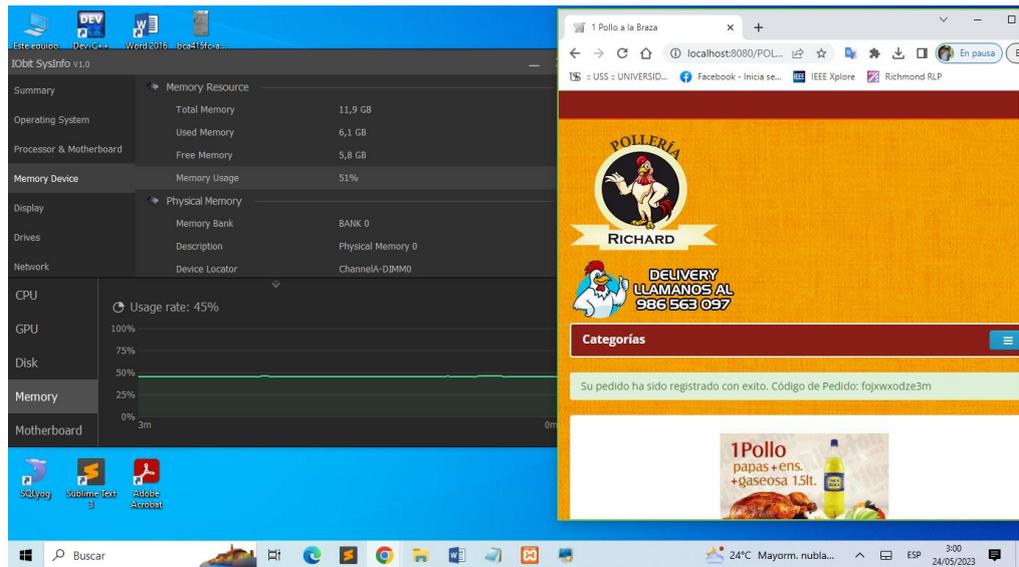


Figura 32: Utilización de Memoria RAM al ejecutar el proceso de Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

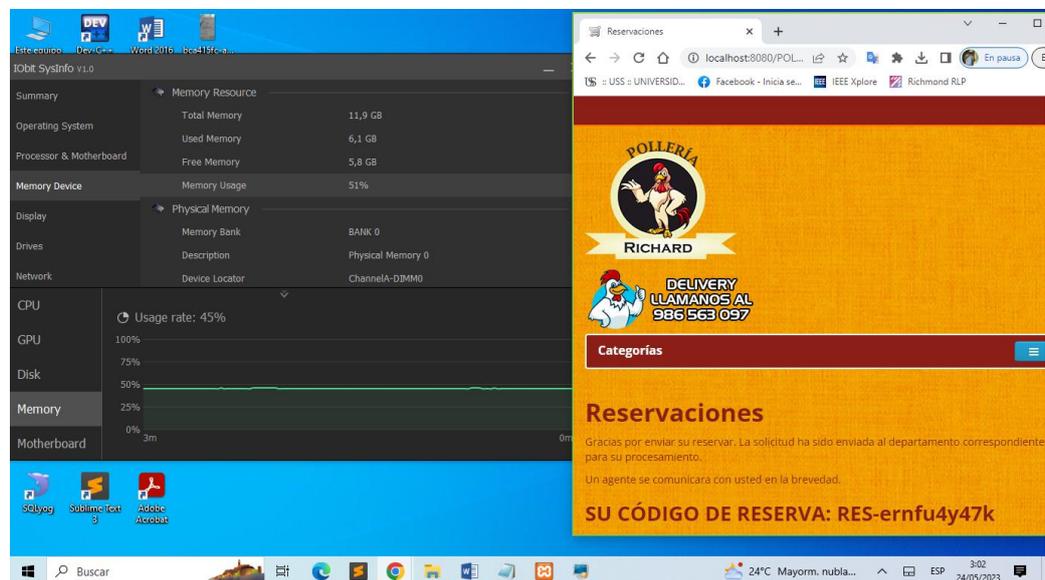


Figura 33: Utilización de Memoria RAM al ejecutar el proceso de Reservaciones web de la aplicación web MVP. Fuente: *Elaboración propia*

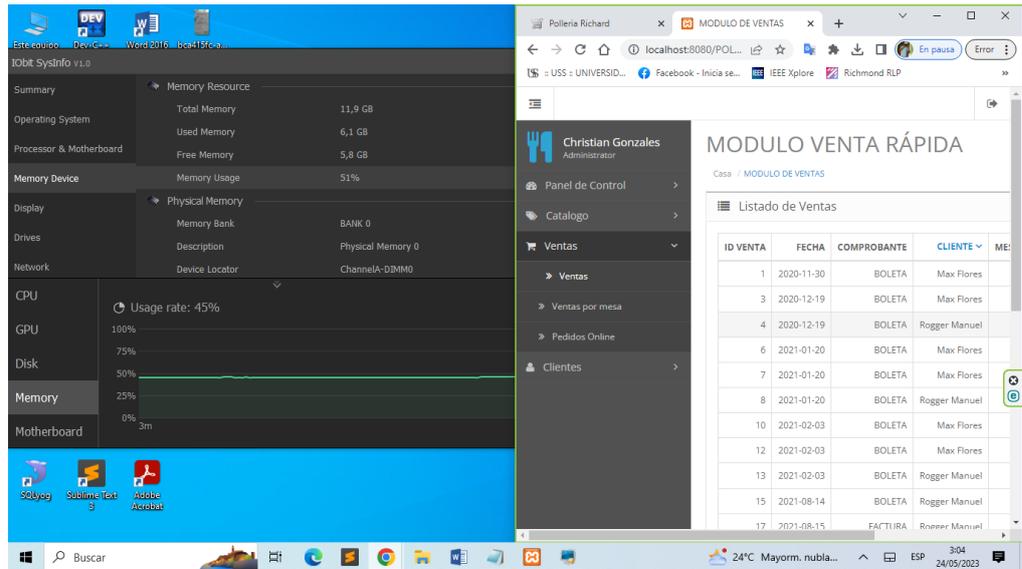


Figura 34: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVP. Fuente: *Elaboración propia*

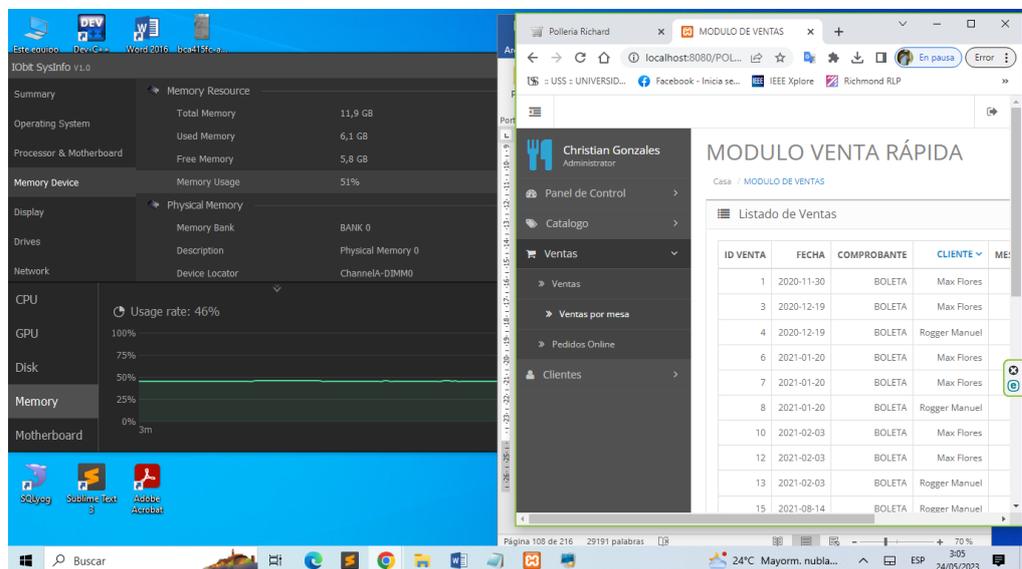


Figura 35: Utilización de Memoria RAM al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVP. Fuente: *Elaboración propia*

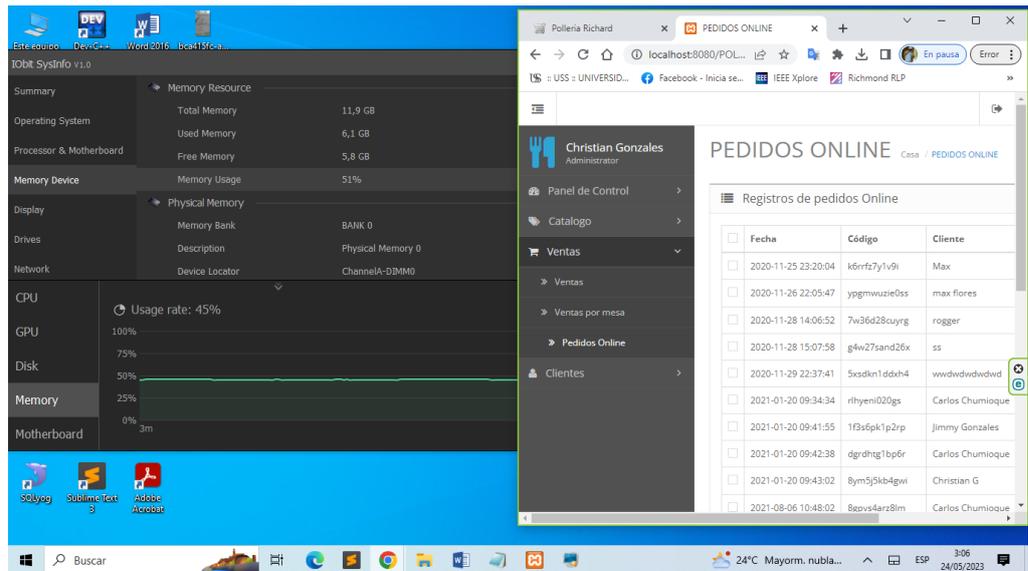


Figura 36: Utilización de Memoria RAM al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

Para calcular el uso de Memoria RAM en el patrón de diseño MVP, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso de la Memoria RAM 5 400 MB del total que tiene la laptop que es de 12 000 MB.

**Total de Memoria RAM utilizada al implementar el Patrón de Diseño MVP: 5 400 MB**

### 3.1.3.2. Uso de CPU

Este indicador tiene referencia a la cantidad de uso de CPU ocupada por la aplicación web al realizar un proceso, el cual cada aplicación esta creada e implementada con cada patrón de diseño.

Para medir este indicador se realizó diferentes pruebas al ejecutar las dos aplicaciones web elaboradas, tiene como objetivo la comparación de las características de rendimiento, en el cual se determinará el uso de CPU al momento de realizar diferentes procesos de la aplicación web.

#### – Escenario 1 para realizar las pruebas

Para realizar las pruebas y recopilación de información, para este indicador se utilizó una computadora con las siguientes características.

Tabla 20:

*Escenario para realizar las pruebas de la Variable Recursos*

<b>Computadora</b>	
<b>Hardware</b>	
<i>Procesador</i>	Intel Pentium D 820
<i>Memoria RAM</i>	1 GB (1 024 MB)
<i>Disco duro</i>	500 GB
<b>Software</b>	
<i>Sistema operativo</i>	Windows 7
<i>Base de datos</i>	Mysql
<i>Servidor local</i>	XAMPP

Fuente: Elaboración propia

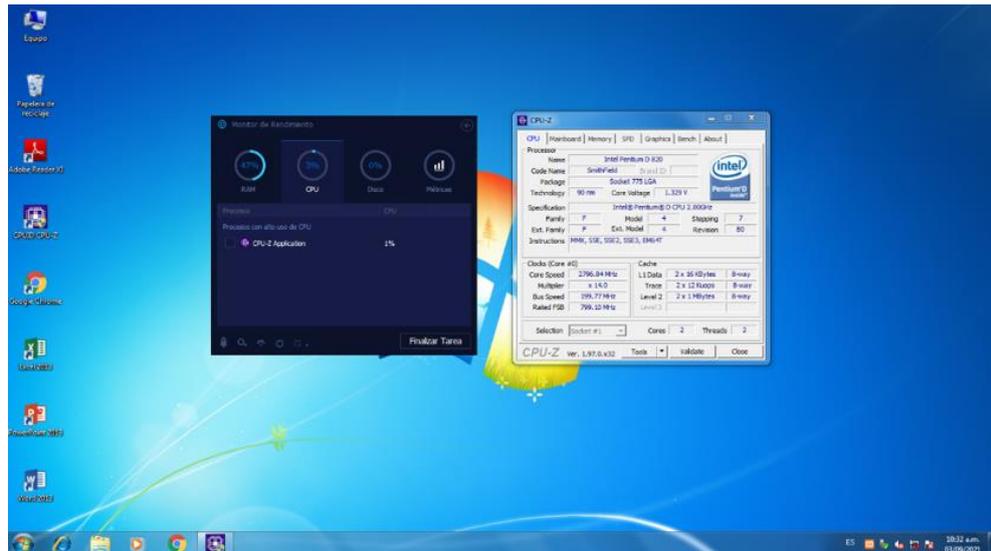


Figura 37: Uso de CPU disponible para la utilización de los patrones de diseño. Fuente: *Elaboración propia*

– **Pruebas realizadas en el Patrón de diseño MVC**

Se realizaron diferentes pruebas para medir el rendimiento en el uso de CPU ejecutando la aplicación web desarrollada con el patrón de diseño MVC.

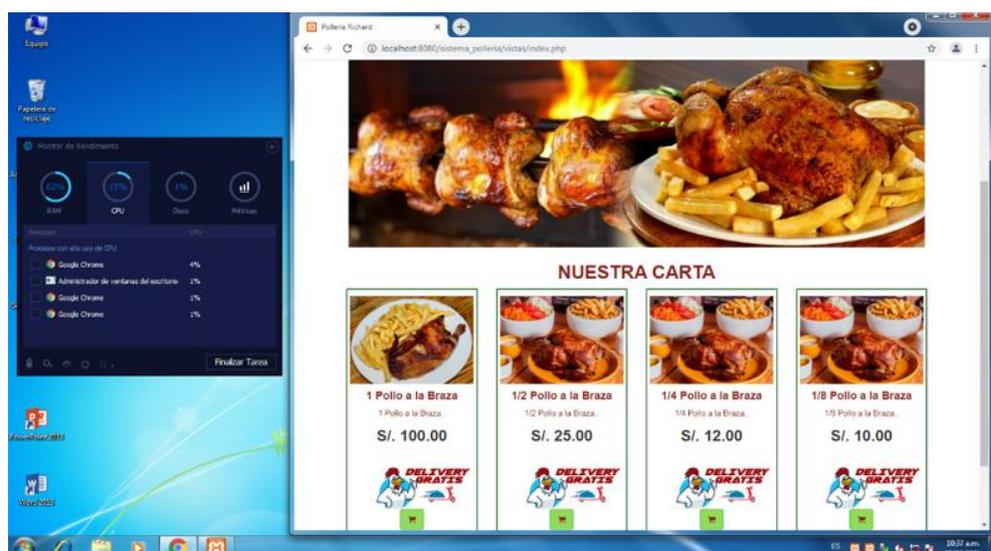


Figura 38: Utilización de Uso de CPU al ejecutar la aplicación web MVC. Fuente: *Elaboración propia*

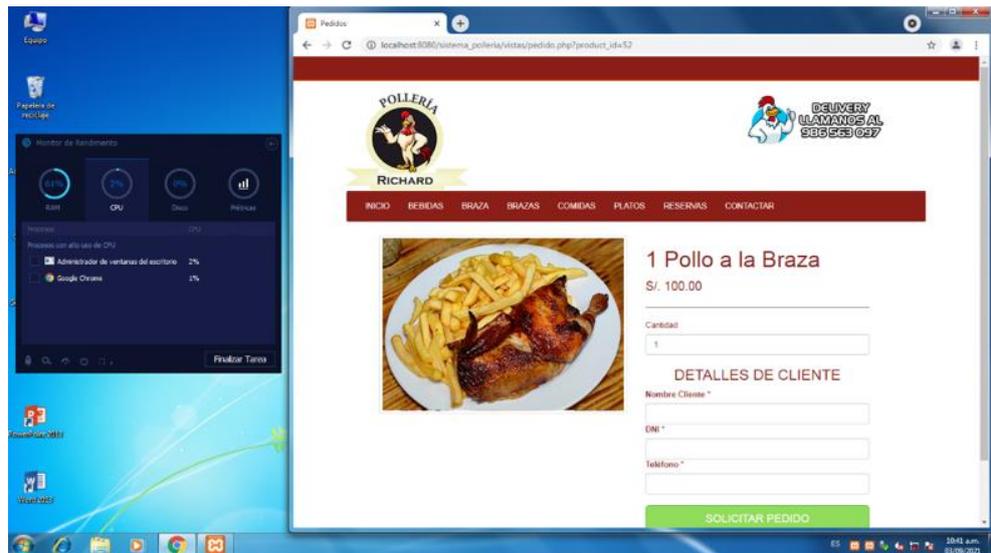


Figura 39: Utilización de Uso de CPU al ejecutar el proceso de Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

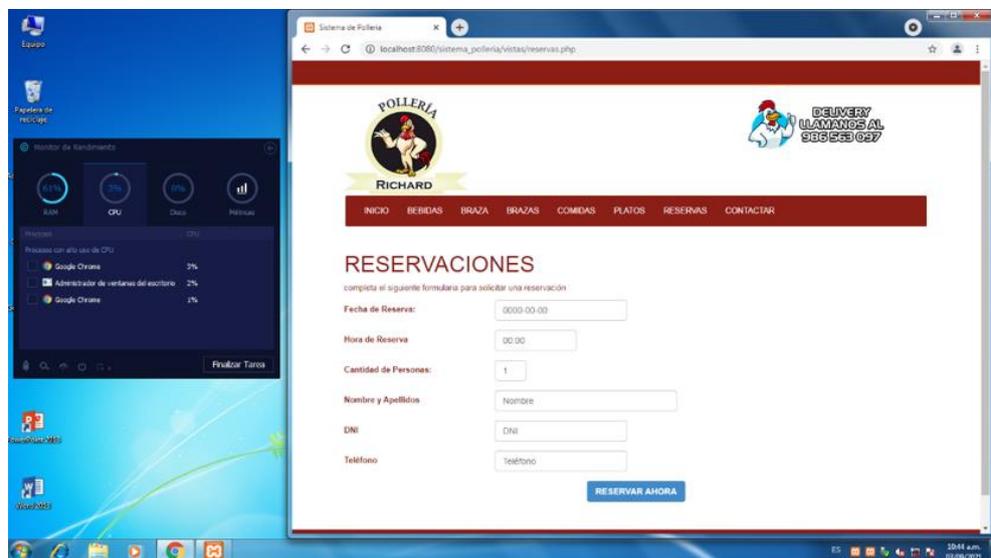


Figura 40: Utilización de Uso de CPU al ejecutar el proceso de Reservas web de la aplicación web MVC. Fuente: *Elaboración propia*

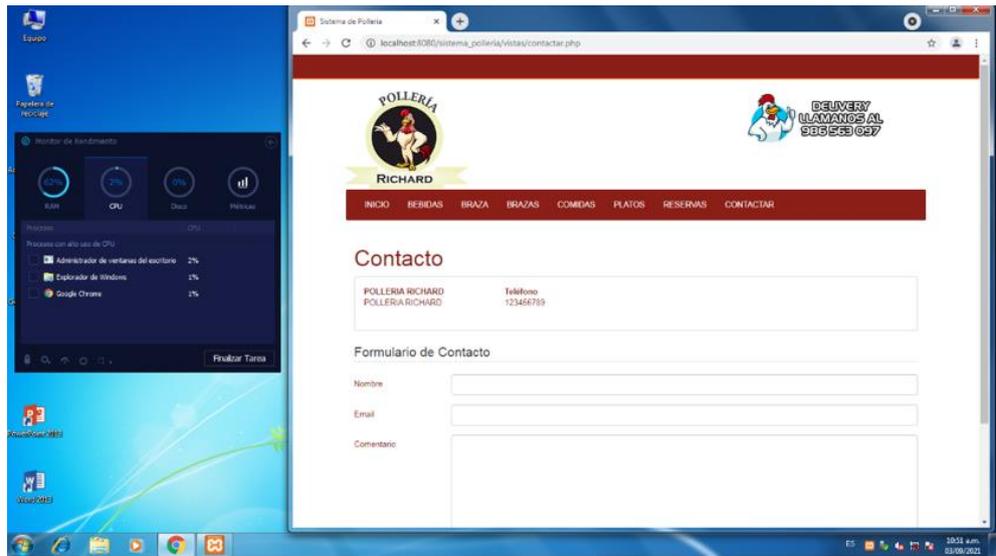


Figura 41: Utilización de Uso de CPU al ejecutar el proceso de Contactar a la pollería de la aplicación web MVC. Fuente: *Elaboración propia*

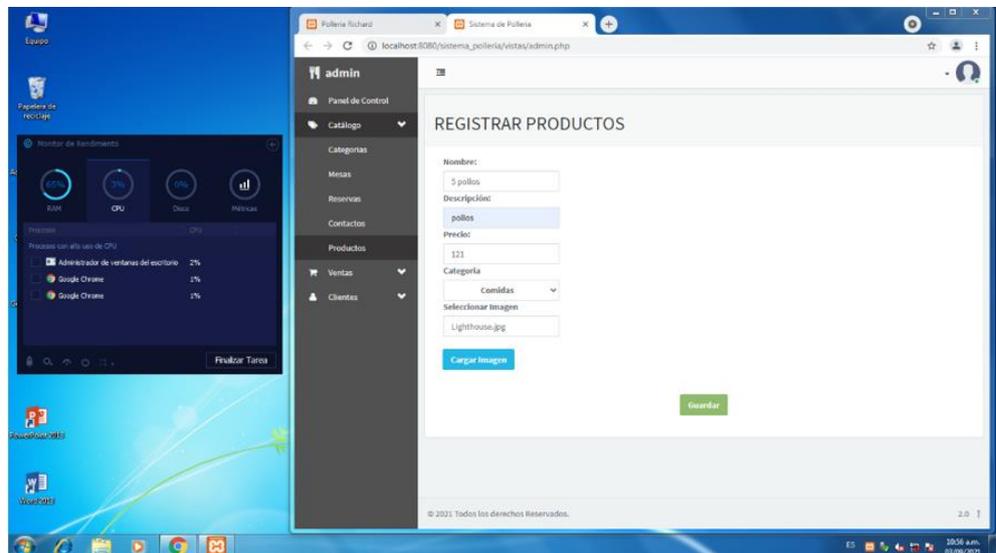


Figura 42: Utilización de Uso de CPU al ejecutar el proceso de Registrar Producto de la aplicación web MVC. Fuente: *Elaboración propia*

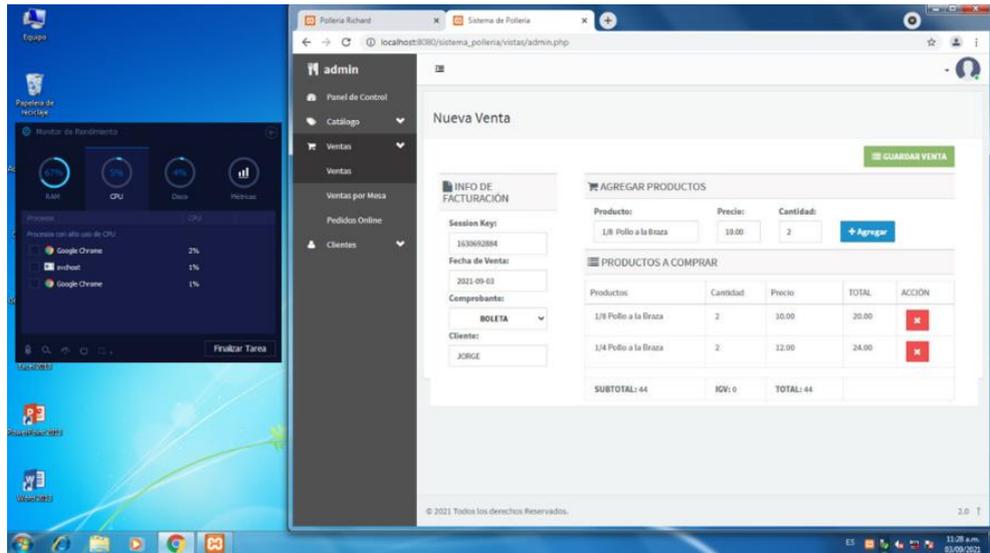


Figura 43: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVC. Fuente: *Elaboración propia*

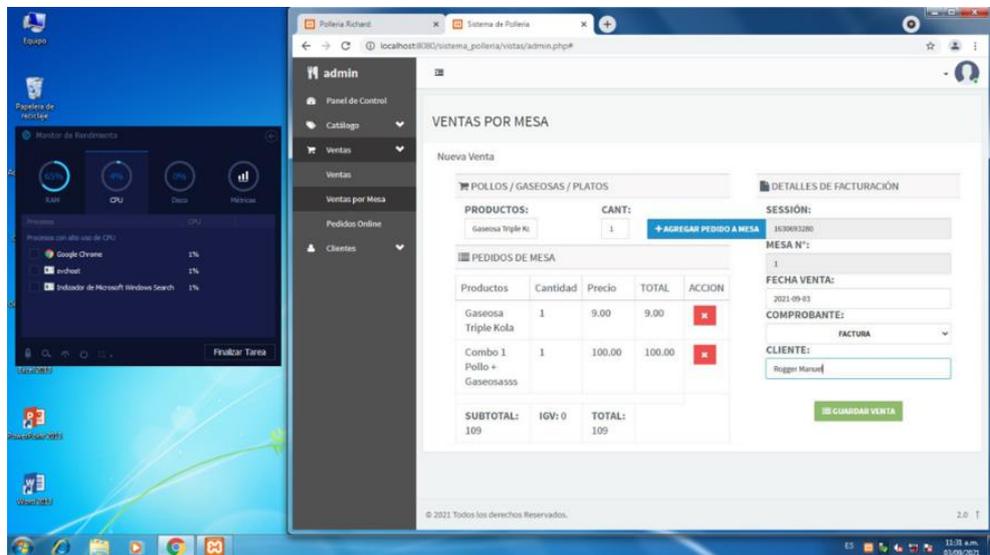


Figura 44: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVC. Fuente: *Elaboración propia*

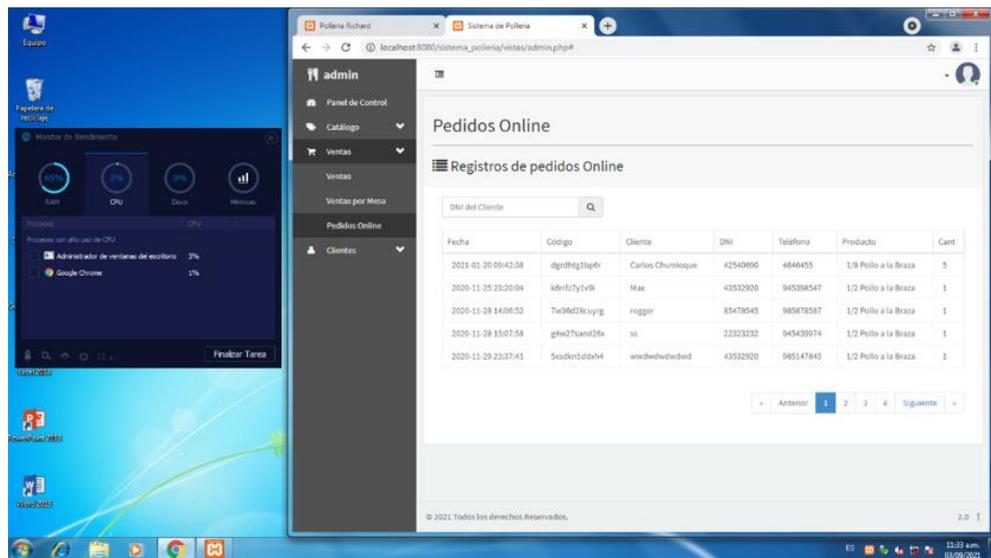


Figura 45: Utilización de Uso de CPU al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

Para calcular el uso de CPU en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso del CPU el 4% total que tiene la computadora.

**Total de proceso ejecutados: 8 procesos**

**Total de Uso de CPU utilizada al implementar el Patrón de Diseño MVC:**

4%

– **Pruebas realizadas en el Patrón de diseño MVP**

Se realizaron diferentes pruebas para medir el rendimiento en el uso de CPU ejecutando la aplicación web desarrollada con el patrón de diseño MVP.

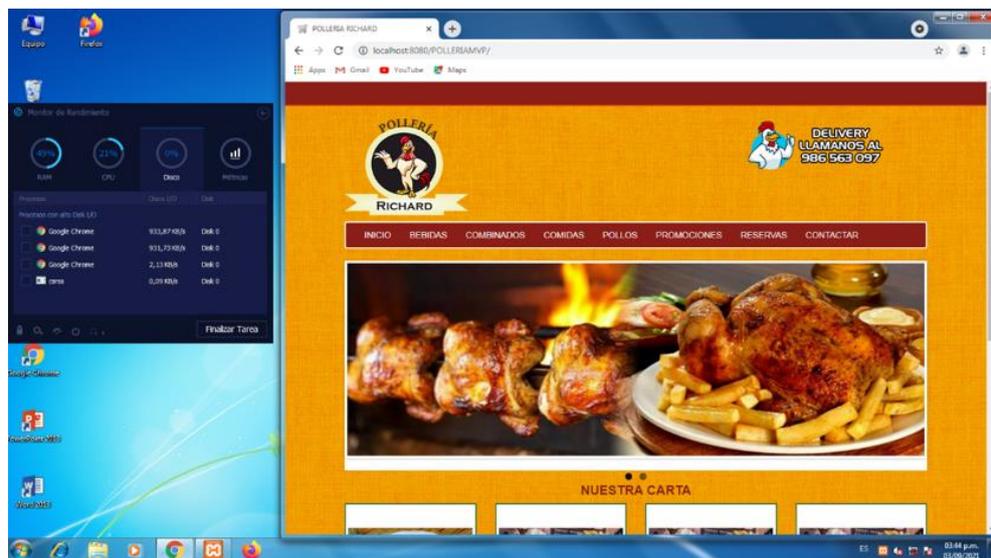


Figura 46: Utilización de Uso de CPU al ejecutar la aplicación web MVP. Fuente: *Elaboración propia*



Figura 47: Utilización de Uso de CPU al ejecutar el proceso de Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*



Figura 48: Utilización de Uso de CPU al ejecutar el proceso de Reservaciones web de la aplicación web MVP. Fuente: *Elaboración propia*

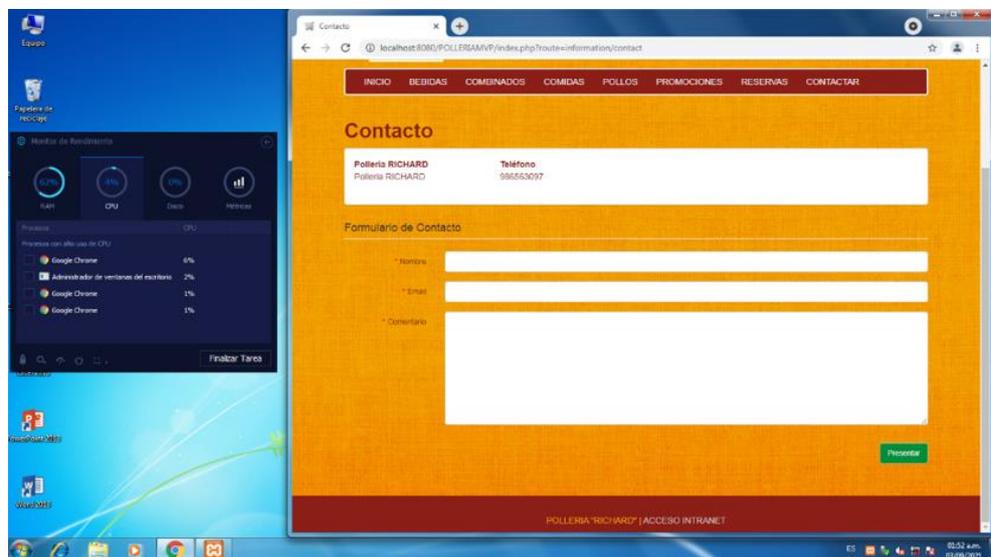


Figura 49: Utilización de Uso de CPU al ejecutar el proceso de Contactar a la pollería de la aplicación web MVP. Fuente: *Elaboración propia*

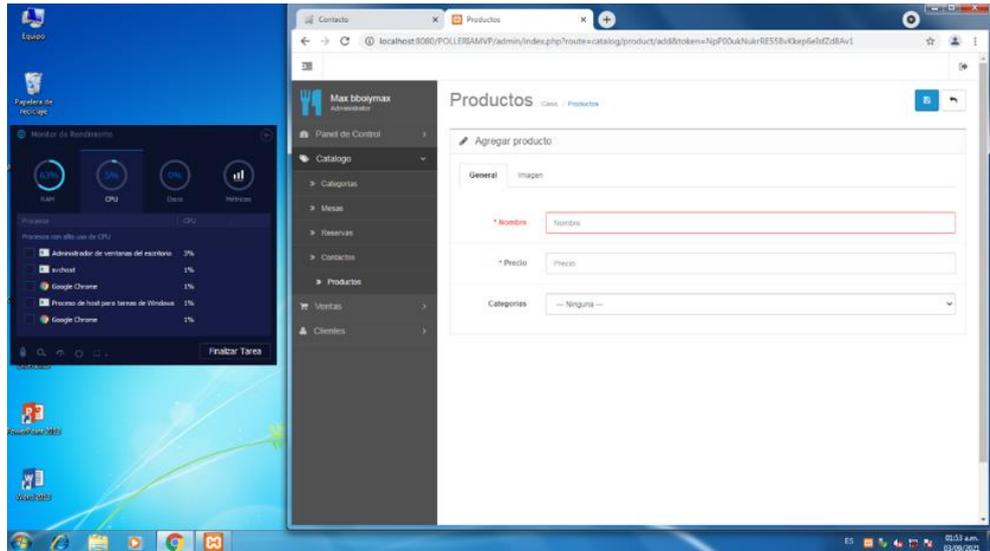


Figura 50: Utilización de Uso de CPU al ejecutar el proceso de Registrar Producto de la aplicación web MVP. Fuente: *Elaboración propia*

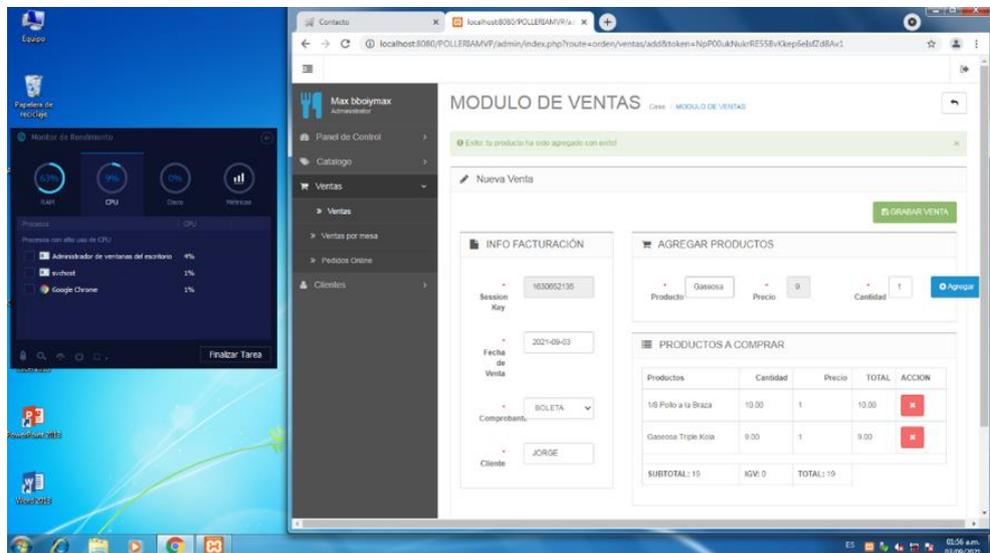


Figura 51: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVP. Fuente: *Elaboración propia*

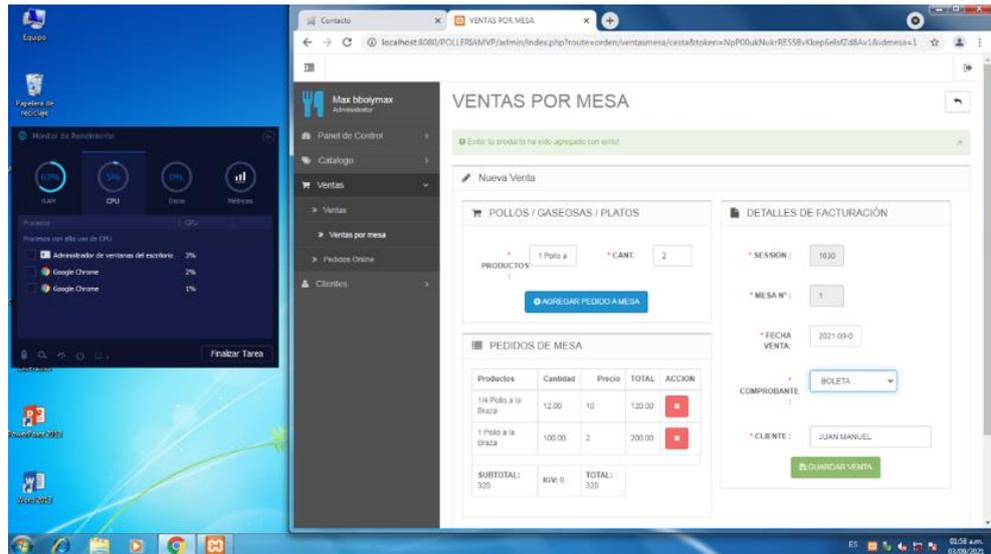


Figura 52: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVP. Fuente: *Elaboración propia*

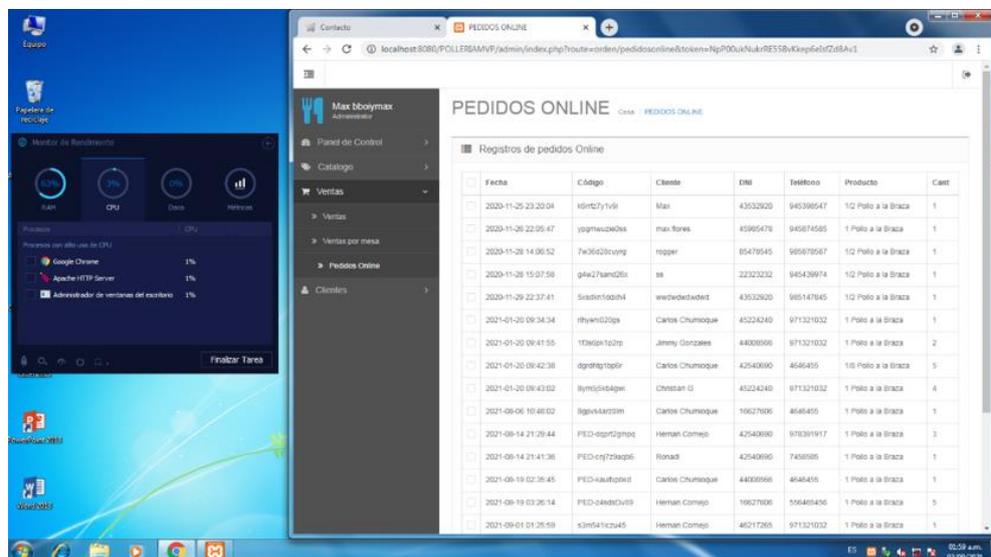


Figura 53: Utilización de Uso de CPU al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

Para calcular el uso de CPU en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso del CPU del 5 % total que tiene la computadora.

**Total de proceso ejecutados:** 8 procesos

**Total de Uso de CPU utilizada al implementar el Patrón de Diseño MVP:**  
5%

– **Interpretación del indicador de Uso de CPU**

*Tabla 21:*

Resultados obtenidos respecto al indicador de uso de CPU

<b>Patrones de diseño</b>	<b>Uso de CPU utilizado</b>	<b>Procesos realizados</b>
Modelo, Vista y Controlador (MVC)	4 %	8
Modelo, Vista y Presentador (MVP)	5 %	8

Fuente: Elaboración propia

– **Escenario 2 para realizar las pruebas**

Tabla 22:

Escenario 2 para realizar las pruebas de la Variable Recursos

<b>Laptop</b>	
<b>Hardware</b>	
Procesador	Intel Core i7-7500U
Memoria RAM	12 GB (12 000 MB)
Disco duro	1 000 GB
<b>Software</b>	
Sistema operativo	Windows 10 Home
Base de datos	Mysql
Servidor local	XAMPP

Fuente: Elaboración propia

– **Pruebas realizadas en el Patrón de diseño MVC**

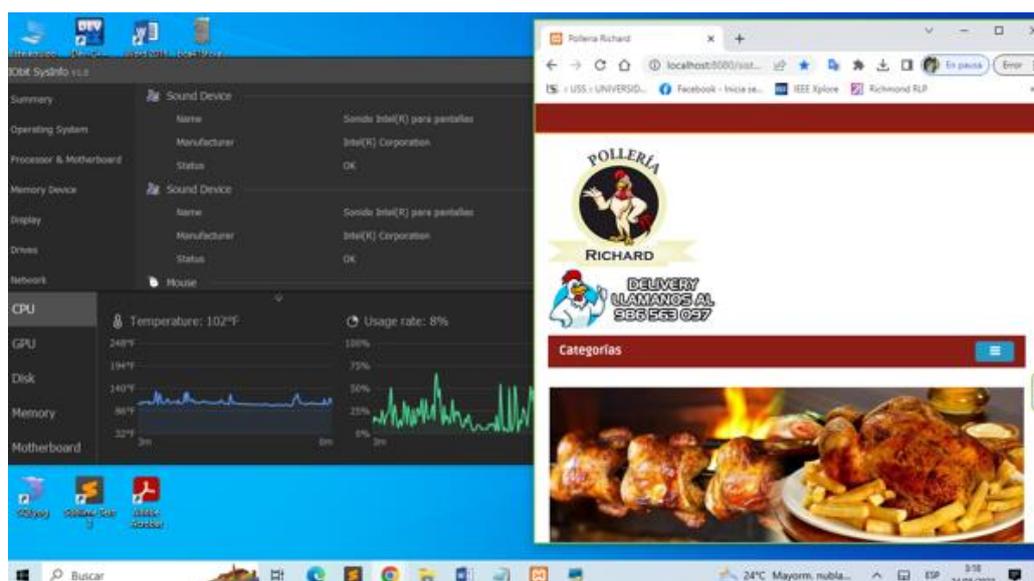


Figura 54: Uso de CPU disponible para la utilización de los patrones de diseño. Fuente: *Elaboración propia*

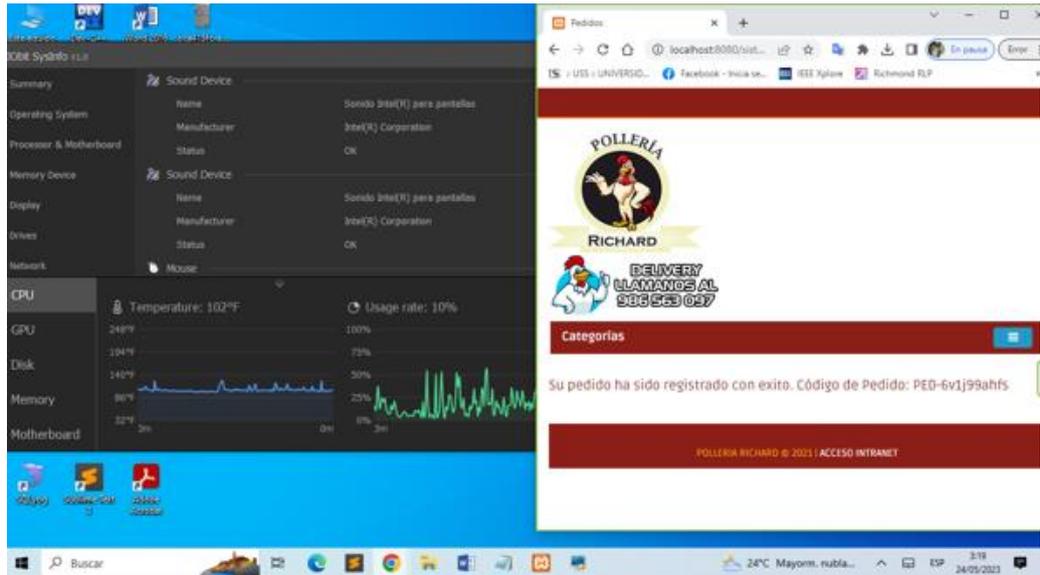


Figura 55: Utilización de Uso de CPU al ejecutar el proceso de Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

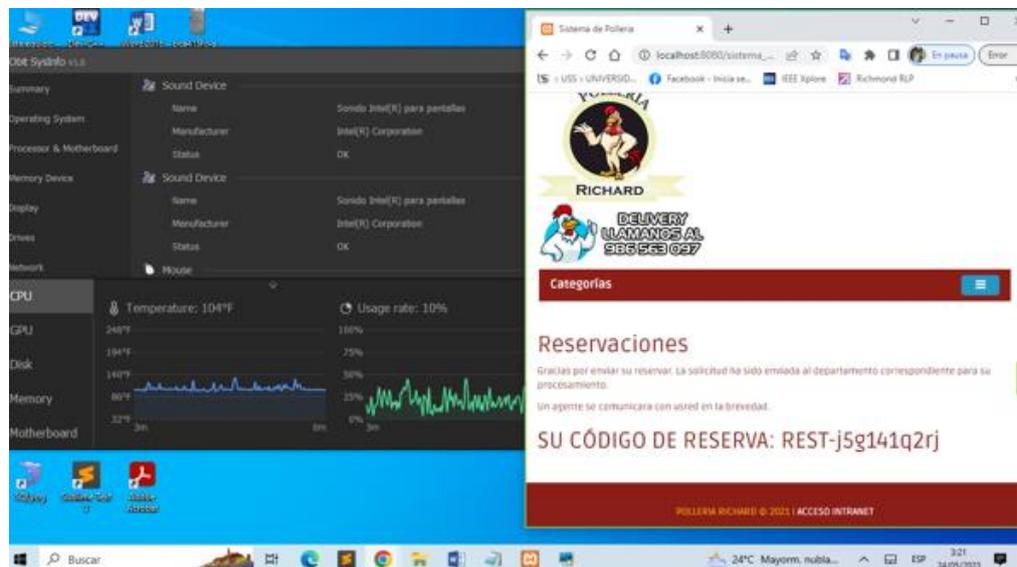


Figura 56: Utilización de Uso de CPU al ejecutar el proceso de Reservaciones web de la aplicación web MVC. Fuente: *Elaboración propia*

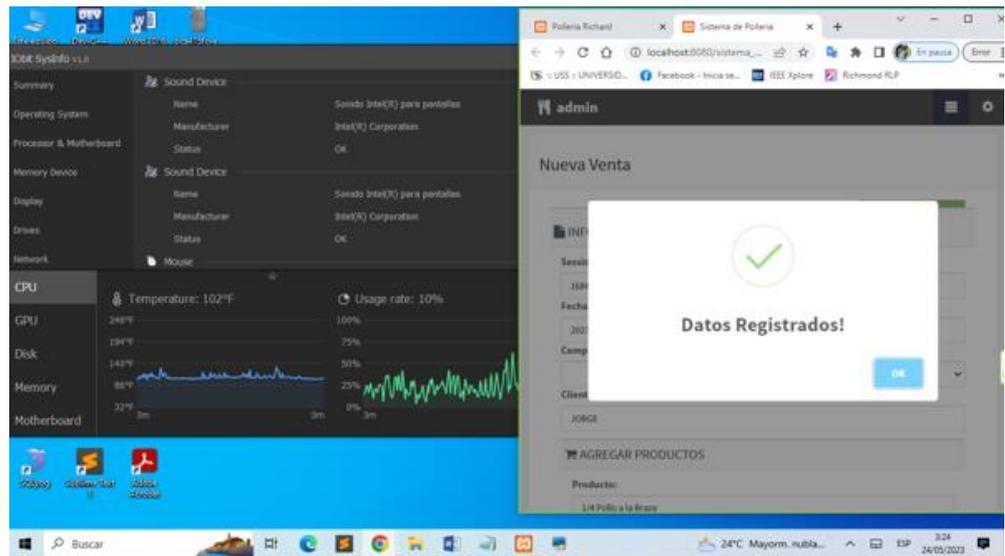


Figura 57: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVC. Fuente: *Elaboración propia*

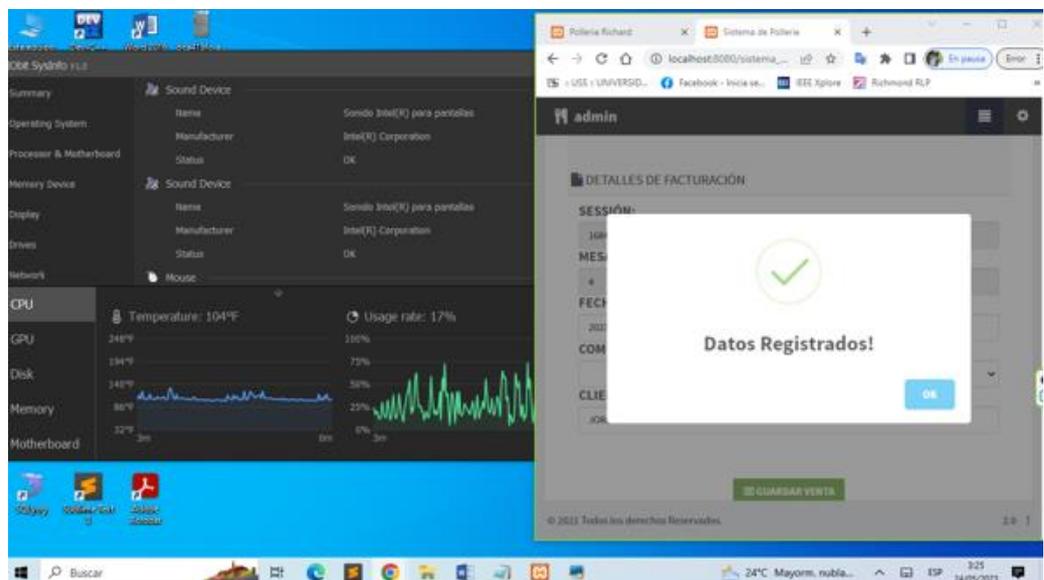


Figura 58: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta por Mesa de la aplicación web MVC. Fuente: *Elaboración propia*

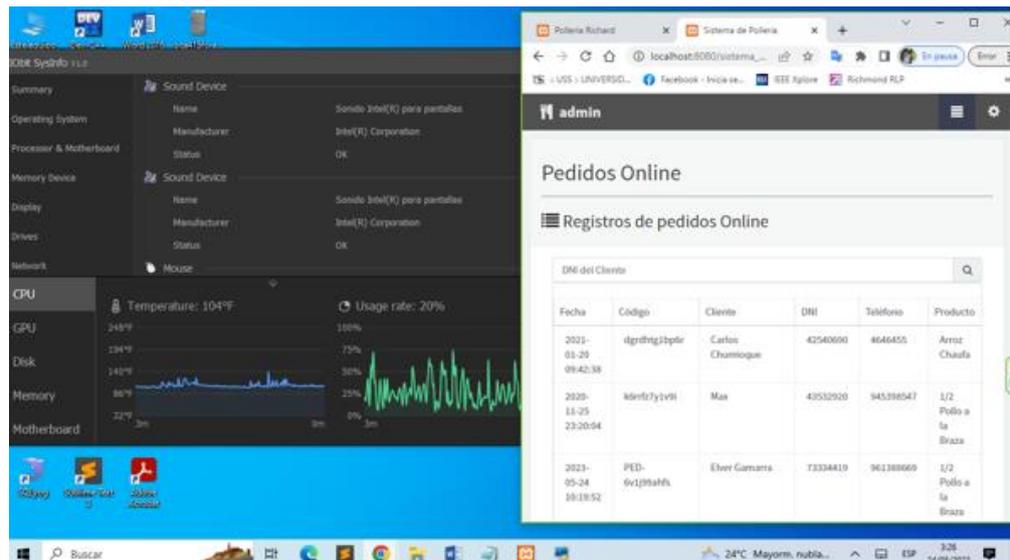


Figura 59: Utilización de Uso de CPU al ejecutar el proceso de Mostrar los Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

Para calcular el uso de CPU en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso del CPU el 13% total que tiene la laptop.

**Total de Uso de CPU utilizada al implementar el Patrón de Diseño MVC:**

13%

– Pruebas realizadas en el Patrón de diseño MVP

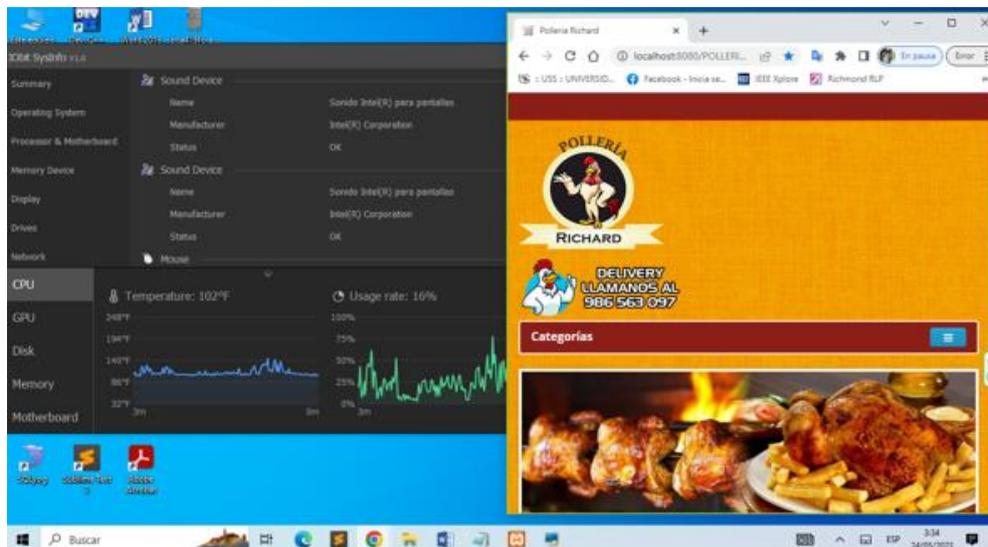


Figura 60: Uso de CPU disponible para la utilización de los patrones de diseño. Fuente: *Elaboración propia*

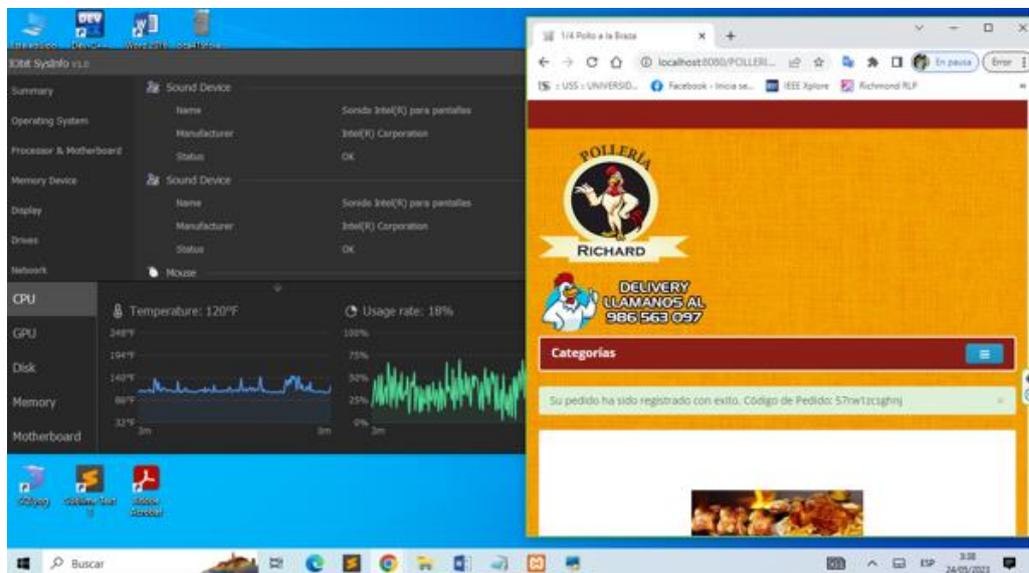


Figura 61: Utilización de Uso de CPU al ejecutar el proceso de Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

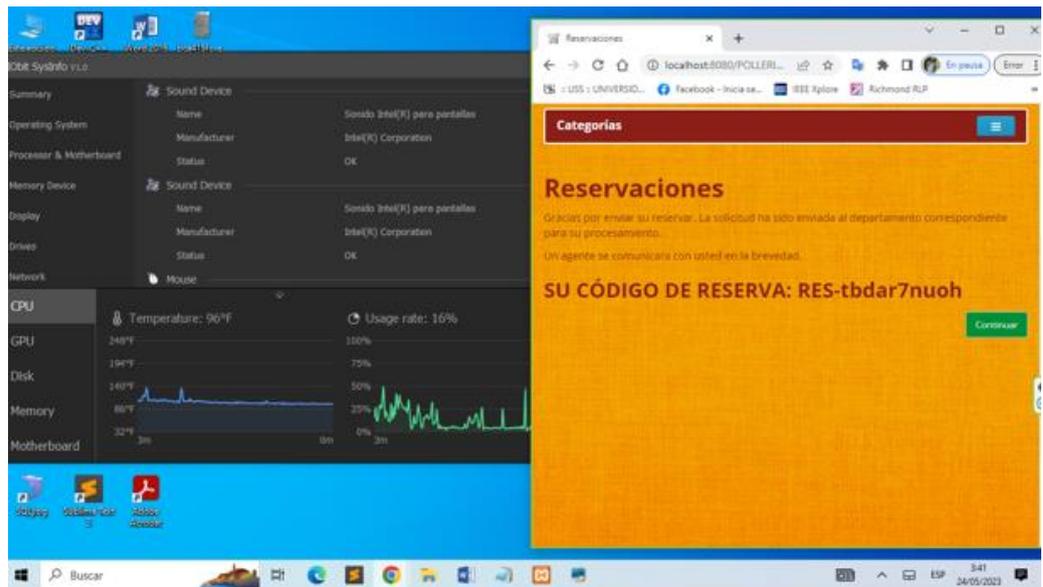


Figura 62: Utilización de Uso de CPU al ejecutar el proceso de Reservaciones web de la aplicación web MVP. Fuente: *Elaboración propia*

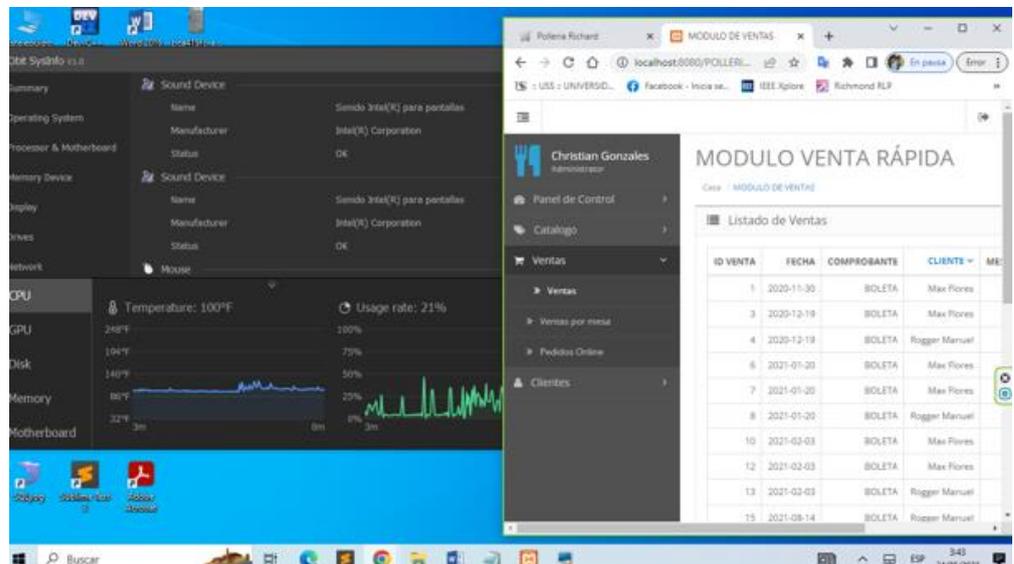


Figura 63: Utilización de Uso de CPU al ejecutar el proceso de Registrar Nueva Venta Directa de la aplicación web MVP. Fuente: *Elaboración propia*



Para calcular el uso de CPU en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de uso del CPU el 18% total que tiene la laptop.

**Total de Uso de CPU utilizada al implementar el Patrón de Diseño MVC:**  
18%

### 3.1.3.3. Tiempo de respuesta

Este indicador se refiere al tiempo que emplea la aplicación web al realizar un proceso, el cual cada aplicación esta creada e implementada con cada patrón de diseño

Para medir este indicador se realizó diferentes pruebas al ejecutar las dos aplicaciones web elaboradas, tiene como objetivo la comparación de las características de rendimiento, en el cual se determinará el tiempo de respuesta al momento de realizar diferentes procesos de la aplicación web.

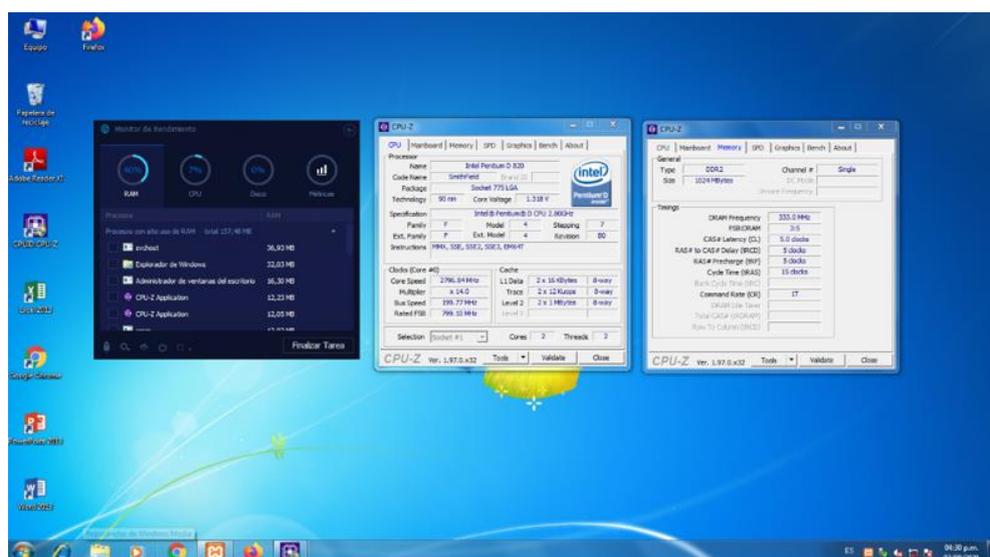


Figura 66: Características de la computadora para las pruebas del indicador Tiempo de Respuesta. Fuente: *Elaboración propia*

- **Herramientas para realizar las pruebas**

Para realizar las pruebas y recopilación de información, para este indicador, se la ha realizado mediante el navegador Google Chrome, en el cual tiene incorporado la opción de medir el tiempo de respuesta al realizar una petición.

- **Pruebas realizadas en el Patrón de diseño MVC**

Se realizaron diferentes pruebas para medir el tiempo de respuesta ejecutando diferentes procesos en la aplicación web desarrollada con el patrón de diseño MVC.

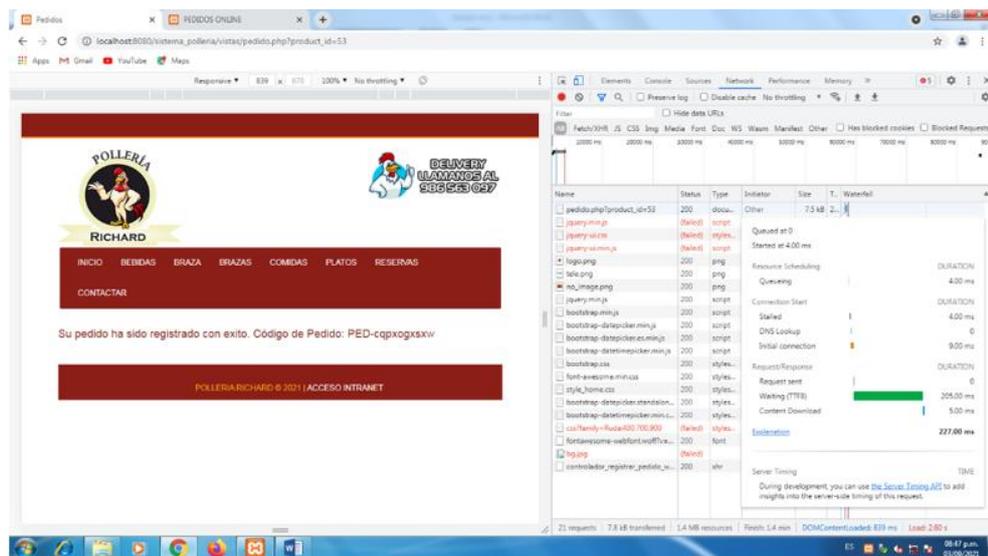


Figura 67: Tiempo de Respuesta al realizar el proceso de Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

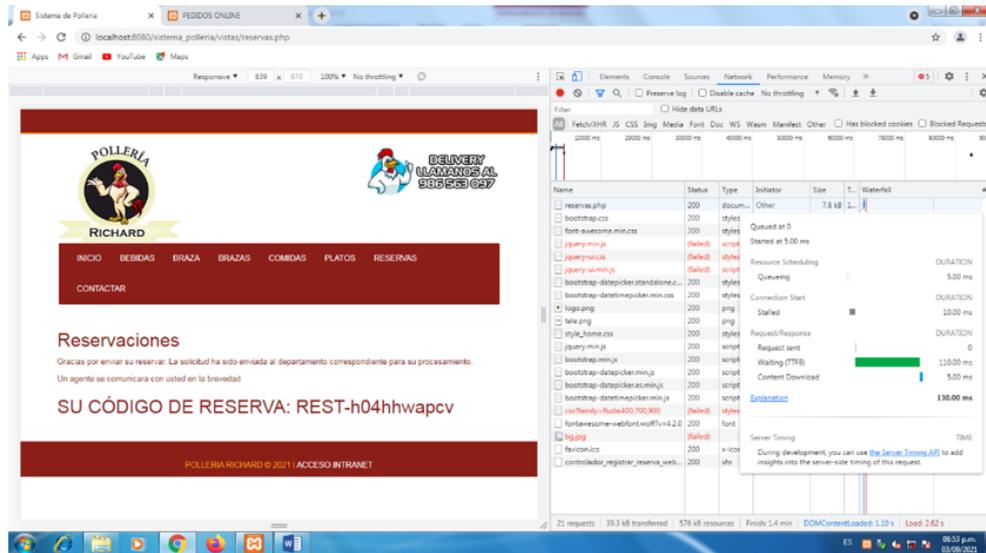


Figura 68: Tiempo de Respuesta al realizar el proceso de Reservaciones Web de la aplicación web MVC. Fuente: *Elaboración propia*

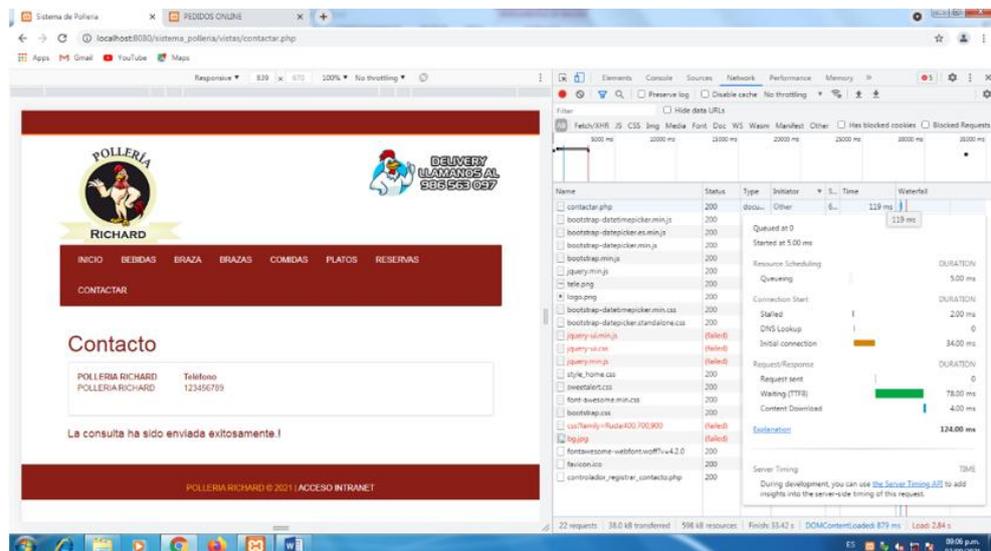


Figura 69: Tiempo de Respuesta al realizar el proceso de Contactar a la pollería de la aplicación web MVC. Fuente: *Elaboración propia*

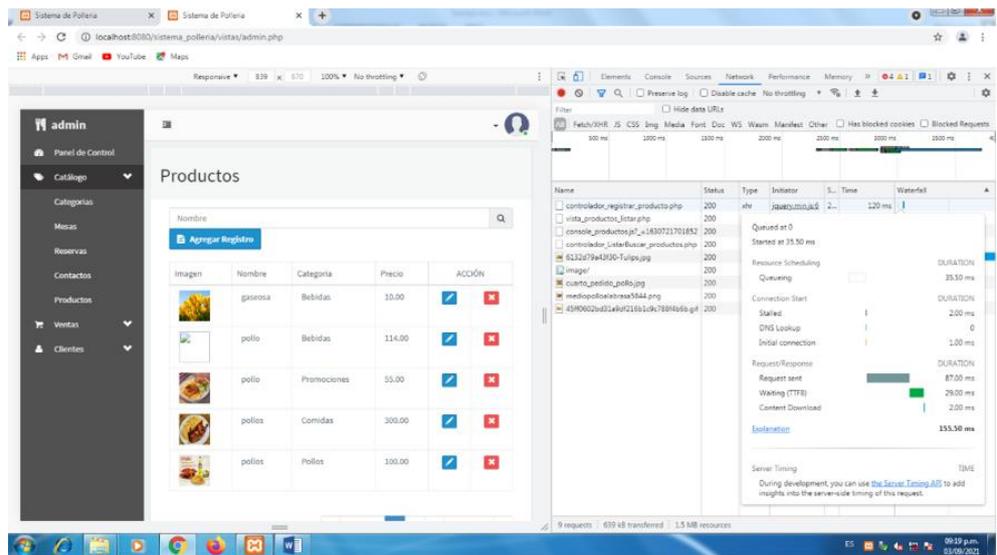


Figura 70: Tiempo de Respuesta al realizar el proceso de Registrar Producto de la aplicación web MVC. Fuente: *Elaboración propia*

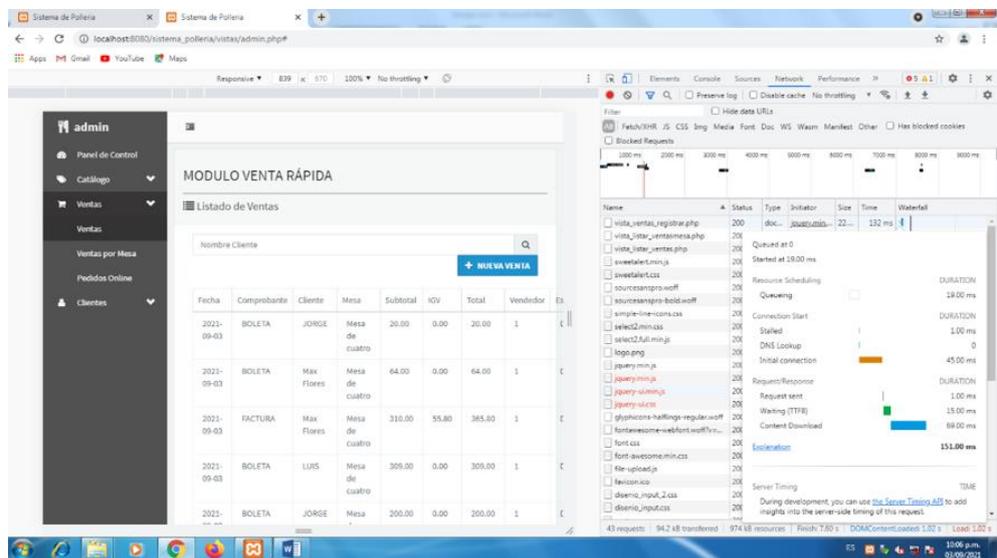


Figura 71: Tiempo de Respuesta al realizar el proceso de Registrar Venta de la aplicación web MVC. Fuente: *Elaboración propia*

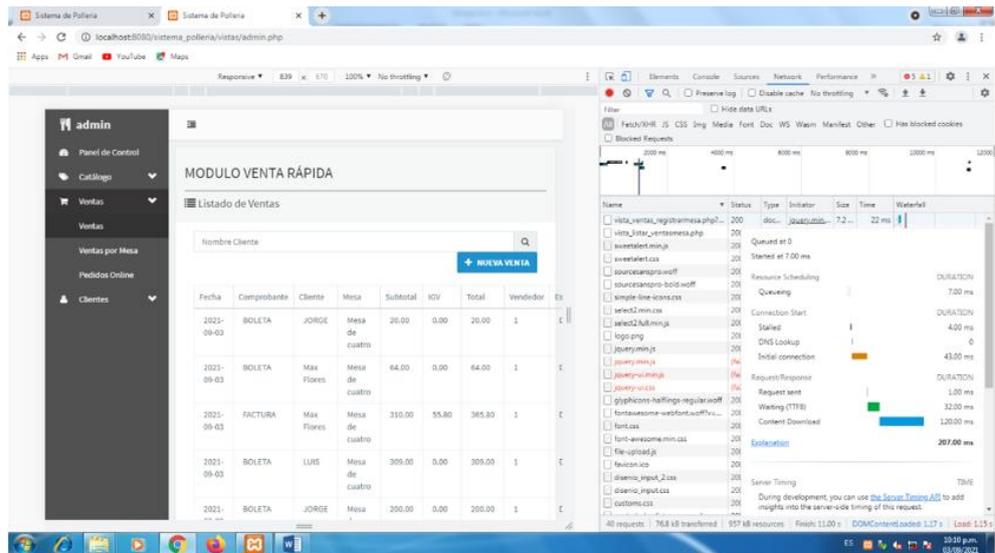


Figura 72: Tiempo de Respuesta al realizar el proceso de Registrar Venta por Mesa de la aplicación web MVC. Fuente: *Elaboración propia*

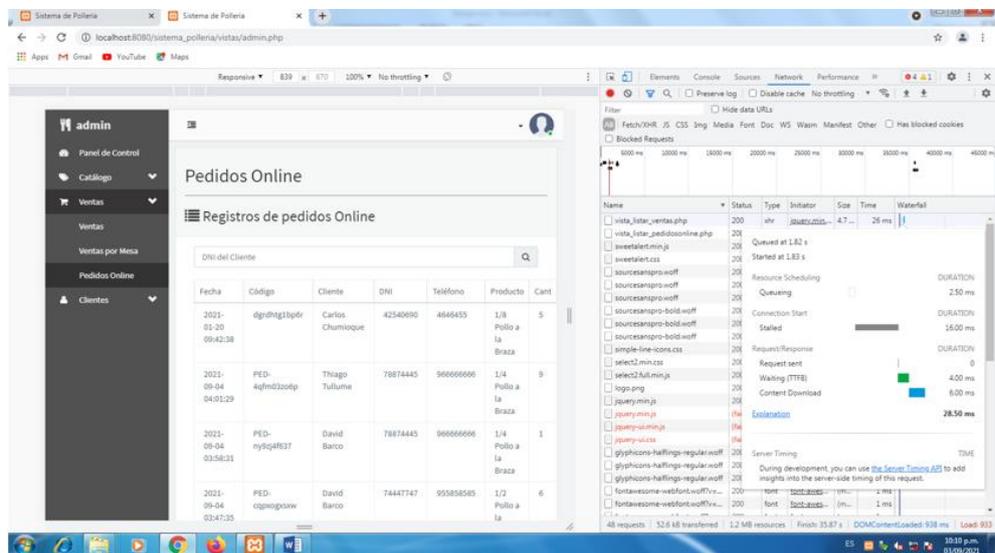


Figura 73: Tiempo de Respuesta al realizar el proceso de Listar los Pedidos Web de la aplicación web MVC. Fuente: *Elaboración propia*

Para calcular el tiempo de respuesta en el patrón de diseño MVC, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de 160,28 milisegundos por cada proceso se realiza en la computadora.

**Total de proceso ejecutados: 7 procesos**

**Total de Tiempo de Respuesta al implementar el Patrón de Diseño**

**MVC: 160,28 milisegundos.**

– **Pruebas realizadas en el Patrón de diseño MVP**

Se realizaron diferentes pruebas para medir el tiempo de respuesta ejecutando diferentes procesos en la aplicación web desarrollada con el patrón de diseño MVP.

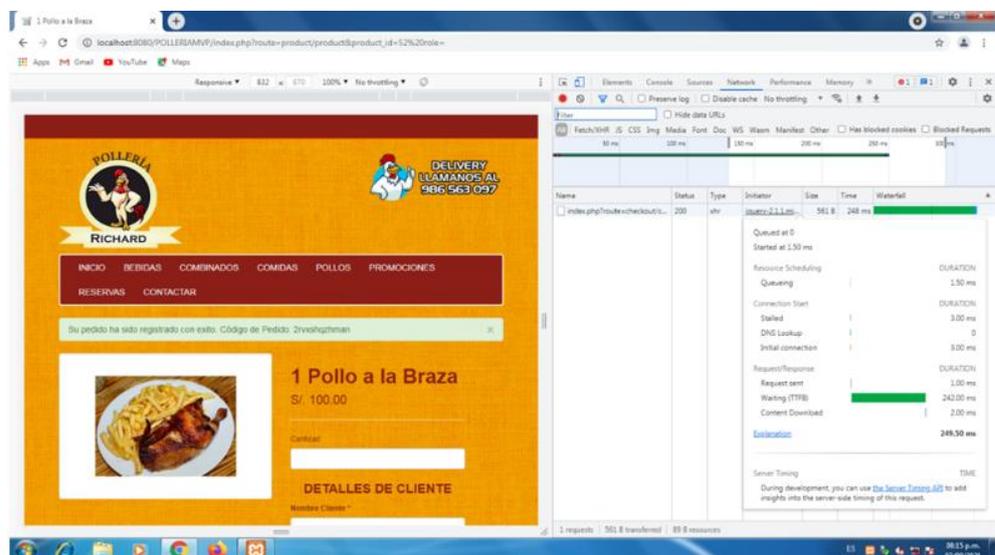


Figura 74: Tiempo de Respuesta al realizar el proceso de Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

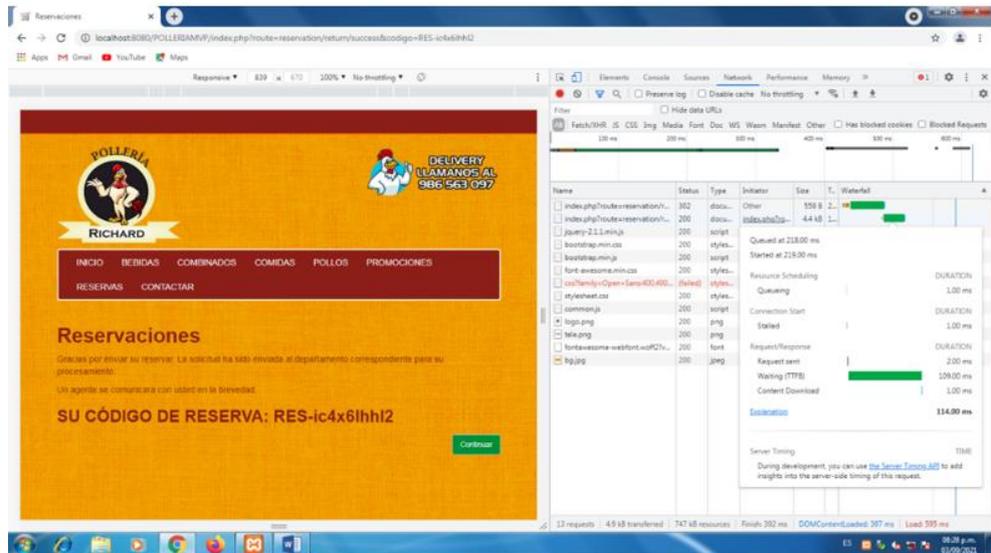


Figura 75: Tiempo de Respuesta al realizar el proceso de Reservaciones Web de la aplicación web MVP. Fuente: *Elaboración propia*

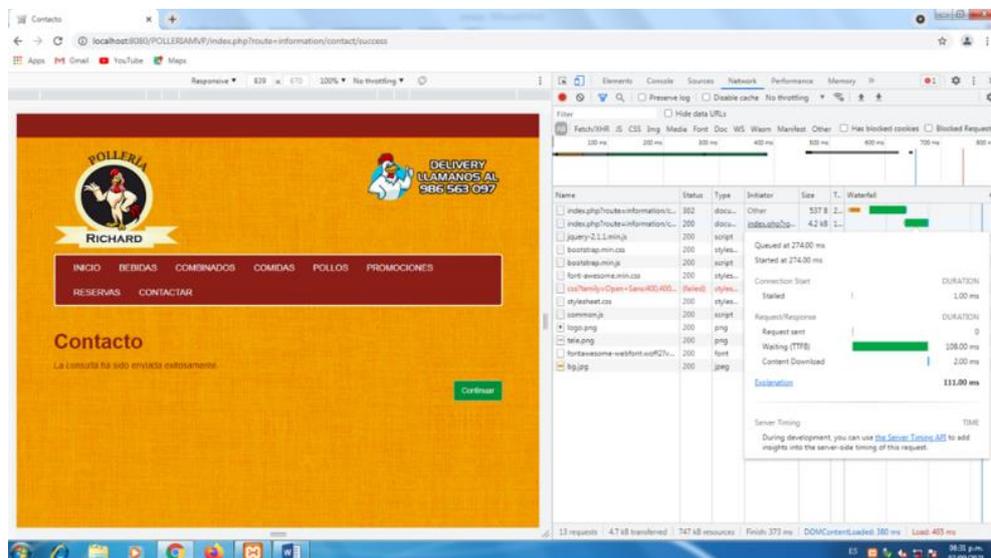


Figura 76: Tiempo de Respuesta al realizar el proceso de Contactar a la pollería de la aplicación web MVP. Fuente: *Elaboración propia*

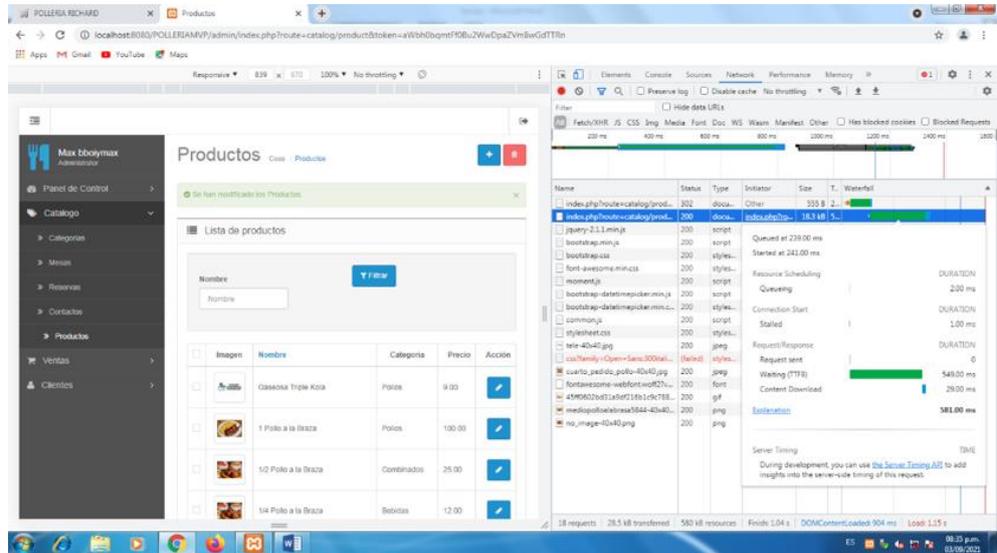


Figura 77: Tiempo de Respuesta al realizar el proceso de Registrar Producto de la aplicación web MVP. Fuente: *Elaboración propia*

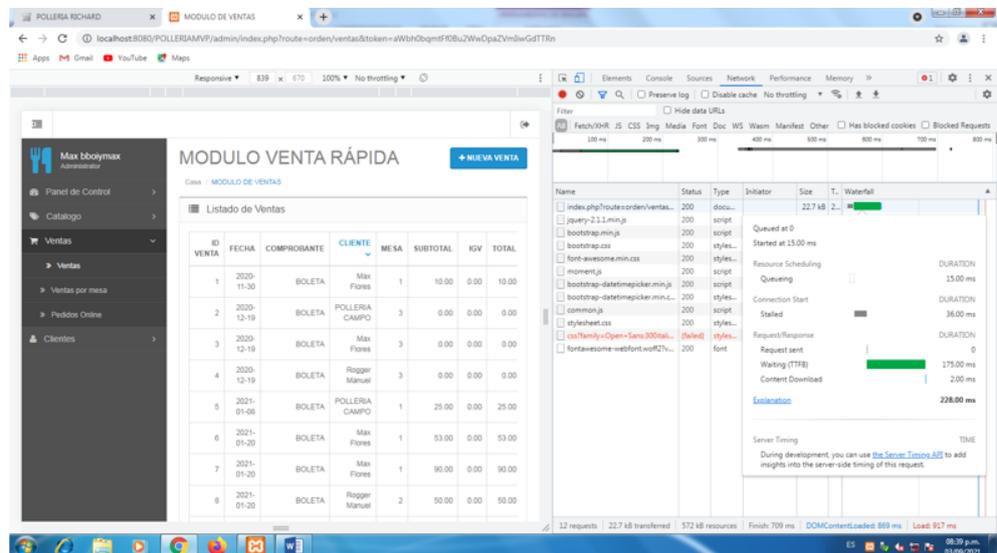


Figura 78: Tiempo de Respuesta al realizar el proceso de Registrar Venta de la aplicación web MVP. Fuente: *Elaboración propia*

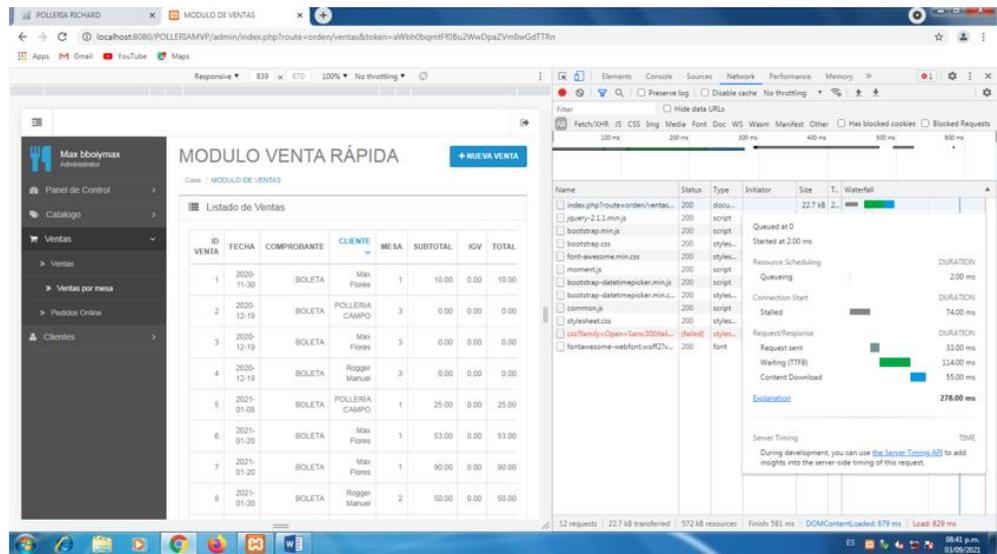


Figura 79: Tiempo de Respuesta al realizar el proceso de Registrar Venta por Mesa de la aplicación web MVP. Fuente: *Elaboración propia*

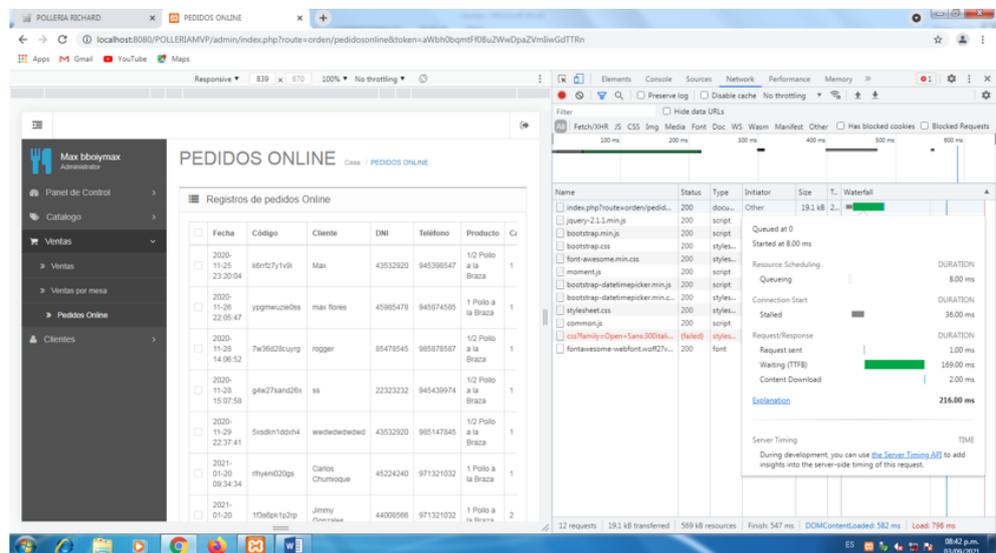


Figura 80: Tiempo de Respuesta al realizar el proceso de Listar los Pedidos Web de la aplicación web MVP. Fuente: *Elaboración propia*

Para calcular el tiempo de respuesta en el patrón de diseño MVP, se calculó ejecutando diferentes procesos que están implementados en la aplicación web, obteniendo un promedio de 253,86 milisegundos por cada proceso se realiza en la computadora.

**Total de proceso ejecutados:** 7 procesos

**Total de Tiempo de Respuesta al implementar el Patrón de Diseño**

**MVP:** 253,86 milisegundos.

– **Interpretación del indicador de Tiempo de Respuesta**

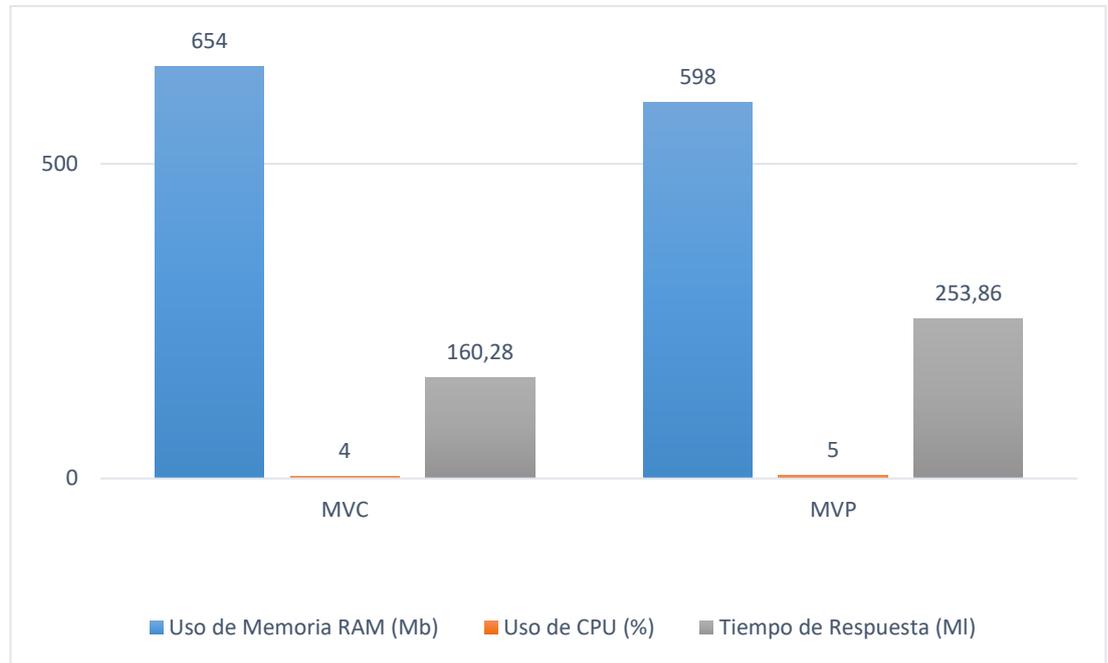
*Tabla 23:*

Resultados obtenidos respecto al indicador Tiempo de Respuesta

<b>Patrones de diseño</b>	<b>Tiempo de Respuesta</b>	<b>Procesos</b>
Modelo, Vista y Controlador (MVC)	160,28 milisegundos	7
Modelo, Vista y Presentador (MVP)	253,86 milisegundos	7

Fuente: Elaboración propia

### 3.1.4. Interpretación de los resultados de la variable Recursos



*Figura 81: Interpretación de la Variable Recursos. Fuente: Elaboración propia*

En la variable Recursos se muestra que el patrón de diseño MVC tiene ciertas ventajas en el tiempo de respuesta con el patrón de diseño MVP al realizar diferentes procesos.

### 3.1.5. Comparación porcentual de los patrones de diseño según sus variables

Tabla 24:

Resultados porcentuales de los patrones de diseño según sus variables

<b>Patrones de diseño</b>	<b>Variable Desarrollo</b>	<b>Variable Recursos</b>
Modelo, Vista y Controlador (MVC)	100 %	100%
Modelo, Vista y Presentador (MVP)	90 %	95%

Fuente: Elaboración propia

En la variable Desarrollo al implementar el patrón de diseño MVC nos conlleva a utilizar menos tiempo de desarrollo con un total de 280 horas e implementar menos líneas de código con un total de 3 098 lo que significa una optimización del 100 %; en cambio al utilizar el patrón de diseño MVP se obtiene el 90 % de optimización ya que en tiempo de desarrollo conlleva a un total de 320 horas e implementar más líneas de código con un total de 7 587 para obtener la misma aplicación web. Se muestra que tiene mayor optimización el patrón de diseño MVC en la variable desarrollo.

En la variable Recursos al implementar el patrón de diseño MVC nos conlleva a utilizar memoria Ram con un promedio de 654 MB de 1024 MB, al utilizar el uso del CPU con un promedio total del 4%, al ejecutar diferentes procesos de la aplicación web se obtuvo como promedio de tiempo de respuesta de 160,28 milisegundo por proceso, lo que significa una optimización del 100 %; en cambio al utilizar el patrón de diseño MVP se obtiene el 95 % de optimización ya que al utilizar memoria Ram con un promedio de 598 MB de 1024 M, al utilizar el uso del CPU se tiene un promedio del 5%, y el promedio del tiempo de respuesta para ejecutar cada proceso es de 253,86 milisegundos. Se muestra que tiene mayor optimización el patrón de diseño MVC en la variable Recursos.

## **3.2. Discusión**

Se ha evidenciado en los resultados de la investigación, que el patrón de diseño MVC, demostró mejor eficacia en el software desarrollado; para una comprensión mejor y alcance de los resultados, se realizó un contraste con los objetivos y resultados que se lograron en otras investigaciones, que abordan el mismo problema; hay que tener en cuenta que estas investigaciones que se reportan en esta sección solucionan parcialmente el problema y de manera teórica en otros casos.

### **3.2.1. Discusión con otras propuestas**

La primera investigación considerada para la discusión fue llevado a cabo (Khaliluzzaman & Chowdhury, 2016), en donde propone un patrón de diseño al cual modifica y mejora al patrón de diseño tradicional MVC al

agregar dos componentes a dicho patrón pero no se comparan patrones ya existentes y no se le realiza una comparación de cada patrón de diseño en el cual sea desarrollado e implementado cada uno en una aplicación web. Sin embargo, la presente investigación realizó la valoración de los patrones de diseños debido a la variedad de framework en los cuales están basados.

Además, la investigación Khaliluzzaman & Chowdhury solo mide el tiempo de respuesta de cada solicitud brindando un mejor rendimiento el patrón de diseño propuesto en comparación al Silex y Slim; en cambio el actual trabajo realiza una investigación empírica que busca determinar la eficacia de cada patrón de diseño establecido, y para lograrlo se desarrolló dos proyectos de desarrollo de software en el que aplica los patrones de diseño seleccionados para cada software, en el cual se obtuvo que el patrón de diseño más eficaz es el Modelo, Vista y Controlador (MVC).

Así mismo, la investigación de (Kazan, Cantürk, & Bastan, 2015), desarrollo una propuesta muy similar a la desarrollada en la presente investigación, en el cual se hizo una evaluación de la efectividad de utilizar patrones de diseño en un proyecto de software comparando un sistema de seguimiento que no está diseñada con patrones de diseño y otra en la cual ya están implementadas dichos patrones de diseño en el sistema. En cambio, en la presente investigación se evaluaron 2 patrones de diseños que son el Modelo, Vista y Controlador (MVC) y el otro que es Modelo, Vista y Presentador (MVP). Además, Kazan, Cantürk, & Bastan aplica la combinación de 3 patrones de diseño para implementarlo en el sistema de seguimiento y así compararlo con el mismo sistema, pero este mismo sistema no está diseñado con ningún patrón de diseño; a diferencia de la presente investigación que se evaluó dos proyectos de software basados

en la Gestión de Ventas y Pedidos de una Pollería, aplicando un patrón de diseño para cada software y así obtener por separado los resultados de cada patrón de diseño. Los resultados obtenidos reportan que, de 10 usuarios con la nueva aplicación con patrones de diseño en número de operaciones, promedio (milisegundo), mínimo (milisegundo), máximo (milisegundos) es de 230, 17, 1, 651 respectivamente comparado a la aplicación que no está diseñado con patrones de diseño es de 100, 581, 5, 7820. Mientras que en el presente trabajo se reportó que no hay mucha diferencia en tiempos de respuesta (milisegundos) al comparar dos softwares que están implementados con patrones de diseño, en el cual a cada software se le realiza los mismos 7 procesos para así medir su tiempo de respuesta que da como resultado que el patrón de diseño MVC tiene un promedio de respuesta de 160,28 milisegundos en cambio el patrón de diseño MVP tiene un promedio de 253,86 milisegundos por cada proceso realizado.

### **3.2.2. Discusión de los resultados de la presente investigación**

Se realizó una explicación y discusión de los resultados obtenidos tras desarrollar la propuesta, se creyó conveniente realizar un análisis en cada indicador, para así tener una mejor comprensión e inmersión de la investigación.

En cuanto a la variable de desarrollo, que contiene 2 indicadores podemos decir que:

En el indicador de tiempo de desarrollo que tomo realizar en la fase de codificación del patrón de diseño MVC empleo 280 horas para generar los

21 requerimientos que necesita la pollería, mientras que en el patrón de diseño MVP empleo 320 horas para generar los mismos requerimientos. Esto debido a que no se encuentra mucha documentación de implementación del patrón de diseño MVP en las aplicaciones web y se toma más tiempo codificarlo y solucionar diferentes errores que se muestran al implementarlo. En cambio, en el patrón de diseño MVC, hay bastante documentación en el cual se muestra su implementación y errores que son comunes al implementarse ya solucionados. Por lo que podemos decir que el patrón de diseño MVC es significativamente más eficaz que el patrón de diseño MVP en cuanto al indicador de tiempo de desarrollo.

En el indicador de líneas de código escritas que tomo realizar en la fase de codificación del patrón de diseño MVC empleo 3 098 líneas de código escritas para generar los 21 requerimientos que necesita la pollería, mientras que en el patrón de diseño MVP empleo 7 587 líneas de código escritas para generar los mismos requerimientos. Esto debido a que en el patrón de diseño MVP el componente Presentador ejecuta la lógica de negocios, al no tener comunicación directa la Vista con el Modelo, el Presentador realiza las operaciones correspondientes en el modelo y el modelo se vuelve a comunicar con el Presentador, lo que permite implementar más líneas de código en el Presentador para realizar diferentes operaciones que el usuario necesita.

En cambio, en el patrón de diseño MVC, existe una comunicación entre todos los componentes de tal manera que, si el usuario realiza una operación en la vista, la Vista la captura esta operación y transfiere los derechos de procesamiento al Controlador, el Controlador ejecuta la lógica comercial relevante por lo que hay dependencias entre componentes y así

el Controlador no tiene mucha implementación de líneas de códigos escritas en la que ejecuta diferentes operaciones debido a que la vista se ejecuta operaciones. Por lo que podemos decir que el patrón de diseño MVC utiliza menos líneas de código escritas que el patrón de diseño MVP en cuanto al indicador de líneas de código escritas.

En el indicador de uso de Memoria RAM, se tiene que el patrón de diseño MVC utiliza un promedio 654 MB (megabit) del total de Memoria RAM que tiene 1024 MB de Memoria RAM la computadora en el cual se realizó las pruebas mediante 8 procesos que se realizaron, mientras que en el patrón de diseño MVP utiliza un promedio de 598 MB de Memoria RAM. Lo cual demostró que el patrón de diseño MVP es significativamente más eficaz que el patrón de diseño MVC en cuanto al indicador de uso de Memoria RAM.

En el indicador de uso de CPU, se tiene que el patrón de diseño MVC utiliza un promedio del 4% del total de uso de CPU que tiene la computadora en el cual se realizó las pruebas mediante 8 procesos que se realizaron, mientras que en el patrón de diseño MVP utiliza un promedio de 5% de Uso de CPU. Lo cual demostró que el patrón de diseño MVC es significativamente más eficaz que el patrón de diseño MVP en cuanto al indicador de uso de CPU.

En el indicador de Tiempo de Respuesta, se tiene que el patrón de diseño MVC al realizar una petición se obtuvo un promedio de tiempo de respuesta de 160,28 milisegundos por proceso del cual fueron 7 procesos realizados, mientras que en el patrón de diseño MVP obtuvo un promedio de tiempo de respuesta de 253,86 milisegundos. Lo cual demostró que el patrón de diseño MVC es significativamente más eficaz que el patrón de diseño MVP en cuanto al indicador de Tiempo de Respuesta.

### **3.3. Aporte de la investigación**

#### **3.3.1. Aporte Practico**

La propuesta de esta investigación, se comenzó con la documentación e identificación de los patrones de diseño de software, así como también la importancia que ofrece la web en el cual podemos leer diferentes informaciones mediante cualquier dispositivo. Luego para establecer los patrones de diseño, se revisó las publicaciones científicas descritas en esta investigación, seguido después de un listado con los patrones de diseño utilizados y los que mejor rendimiento ofrecieron en sus respectivas investigaciones.

Consecuentemente se obtuvieron las puntuaciones y se seleccionaron los patrones de diseño que obtuvieron la mayor valoración para ser evaluadas. Luego, se implementó dos proyectos de software, con la finalidad de aplicar cada patrón de diseño seleccionado en cada software. Finalmente se determinó cuantitativamente el desempeño de los patrones de diseño seleccionados, evidenciando el tiempo de desarrollo, líneas de código escritas, uso de memoria RAM, uso de CPU y tiempo de respuesta.

En la figura siguiente, se muestra mediante el flujograma la propuesta de investigación. En la imagen detalladamente se muestra paso a paso y se empieza con la identificación de los patrones de diseño existentes hasta llegar a determinar cuál de los patrones de diseño muestra más eficacia según los indicadores.

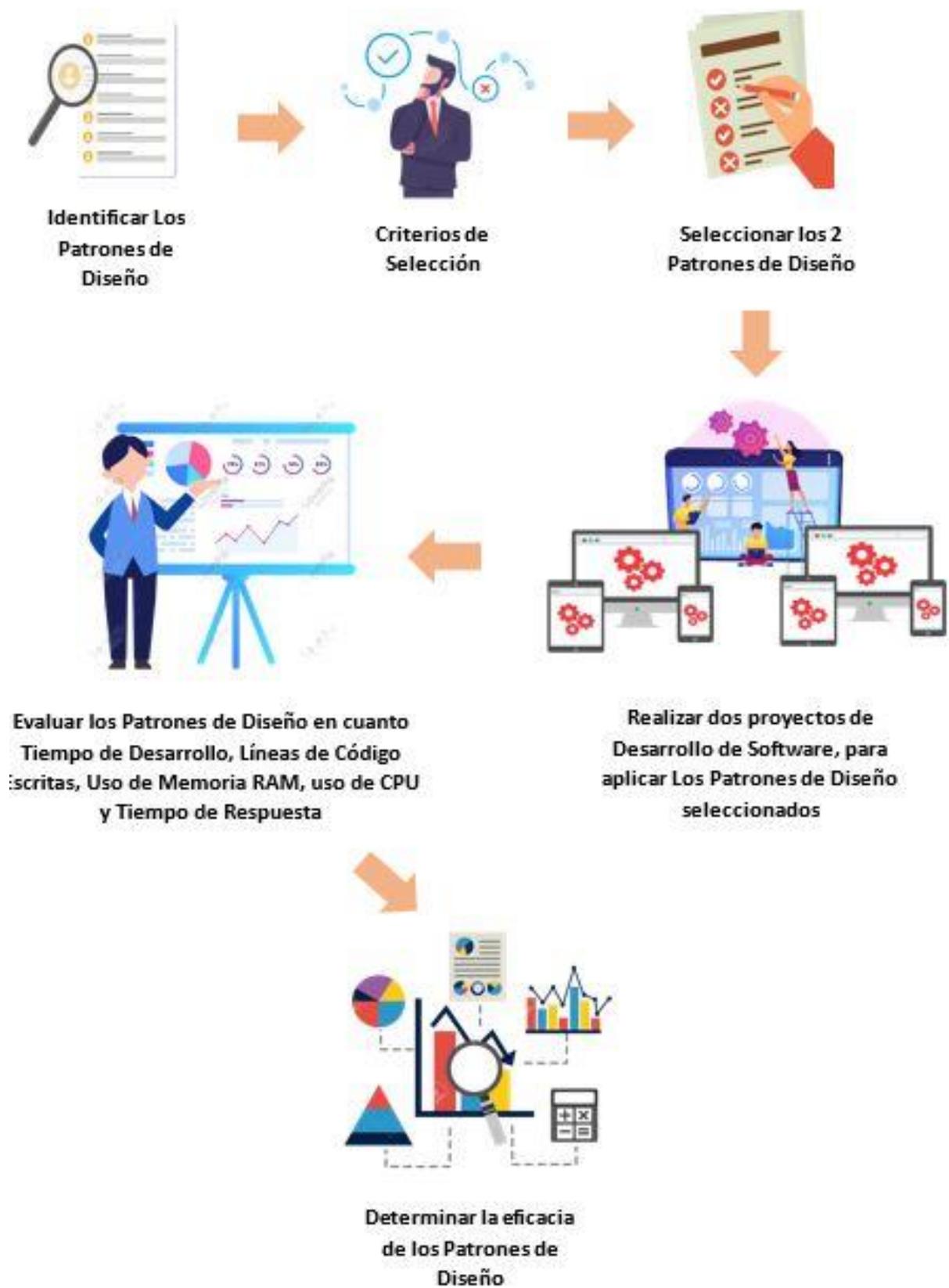


Figura 82: Flujo del proceso que se sigue en la presente investigación. Fuente: Elaboración propia

Como se ha documentado previamente en esta investigación, los patrones de diseño son variados, y cada uno tiene su propia implementación. Hacemos referencia en el marco teórico a varios autores, en ellos encontramos (Montero, Zarronadía, & Díaz, 2011); y a (Honrubia López, 2014) que nos dice que para ser considerado patrones deben brindar una solución que resuelva un problema determinado y que se clasifica según una búsqueda de solución a un problema concreto. En este orden de ideas, para empezar este trabajo de investigación, se realizó la identificación y documentación los patrones de diseño.

### **3.3.2. Seleccionar dos Patrones de Diseño**

Existen varios tipos de patrones de diseño de software para reutilizar e implementarlo en el desarrollo del software.

Tabla 25.

*Descripción y Análisis de los Patrones de Diseño*

<b>Cita</b>	<b>Proyecto</b>	<b>Patrón de diseño</b>	<b>Patrones de diseño de Arquitectura</b>	<b>Criterios del Rendimiento</b>	<b>Evaluación del Rendimiento</b>
<b>(Alexander Katzmaier &amp; Martin Hanneghan , 2013)</b>	Evaluación de patrones de diseño en marcos de aplicaciones móviles y basadas en web.	<b>Modelo, Vista y Controlador (MVC)</b>	<b>Si</b>	Implementación de cada patrón, soporte al implementarlo, estructura de código deficiente, uso de herramientas adicionales para implementarlo.	<b>Top 1</b>
		Contexts and Dependency Injection (CDI)	No		Top 2
		Front Controller	No		Top 3
<b>(Kazan, Cantürk, &amp; Bastan, 2015)</b>	Análisis de rendimiento de un software desarrollado con y sin patrones de diseño: un estudio de caso	<b>Modelo, Vista y Controlador (MVC)</b>	<b>Si</b>	Tiempo en realizar operaciones los usuarios como: Tiempo mínimo, tiempo máximo, tiempo total y número de operaciones.	<b>Top 1</b>
		<b>Modelo, Vista y Presentador (MVP)</b>	<b>Si</b>		
		Proxy	No		
		Sin Patrones de Diseño	-		Top 2

<b>(Thung, Ng, Thung, &amp; Shahida Sulaiman , 2010)</b>	Mejora de una aplicación web utilizando patrones de diseño: un estudio de caso	<b>Modelo, Vista y Controlador (MVC)</b>	Si	Usabilidad, mantenible para los desarrolladores.	<b>Top 1</b>
		<b>Presentación, Abstracción y Control (PAC)</b>	Si		
		Patrones de Navegación	No		
		Sin Patrones de Diseño	-	Top 2	
<b>(Ouyang, Cao, Wu, Gan, &amp; Wang, 2019)</b>	Diseño de marco de software de sistema de perforación inteligente y arquitectura de datos basada en patrón MVC	<b>Modelo principal, Modelo auxiliar, Vista y Controlador (MMVC)</b>	Si	Es mantenible, mas independencia y mejor escalabilidad	<b>Top 1</b>
<b>(Sheng &amp; Wang, 2018)</b>	El diseño de software de un instrumento modular basado en un patrón proxy	<b>Patrón de diseño Proxy Modificado</b>	No	Es eficiente en el empaquetamiento de datos y reducción de código.	<b>Top 1</b>

<b>(Sharma &amp; Davuluri, 2019)</b>	Patrones de diseño para aplicaciones de aprendizaje automático	<b>Patrón de diseño Composite</b>	<b>No</b>	Son utilizables y eficientes por su característica	<b>Top 1</b>
		<b>Modelo, Vista y Controlador (MVC)</b>	<b>Si</b>		<b>Top 1</b>
<b>(XiaoLong Li, y otros, 2015)</b>	Aplicación del patrón de diseño MVVM en MES (Sistema de Ejecución de Manufactura)	<b>Modelo, Vista y VistaModelo (MVVM)</b>	<b>Si</b>	Es mantenible, facilidad al desarrollarlo, reducción de cantidad de código	<b>Top 1</b>
<b>(Khaliluzzaman &amp; Chowdhury, 2016)</b>	Arquitectura MVC basada en Pre y Post Controller para aplicaciones Web	<b>Precontrolador, Modelo, Vista, Controlador, Postcontrolador (MVC extendible)</b>	<b>Si</b>	Tiempo que demora la solicitud al realizar operaciones	<b>Top 1</b>
		Silex	No		Top 2
		Slim	No		Top 3

<b>(Yingda, Jianzhuang, &amp; Xiaowen, 2018)</b>	Patrón de comando de parámetros variádicos y la utilización en MVC	<b>MVC Variádico</b>	<b>Si</b>	Es eficiente al evitar redundancia	<b>Top 1</b>
<b>(Yehia Elshater, Patrick Martin , &amp; Ehab Hassanein, 2015)</b>	Uso de patrones de diseño para mejorar el rendimiento del servicio web	Patrón Singleton	No	Tiempo de respuesta, número de solicitudes cumplidas por unidad de tiempo.	<b>Top 1</b>
		Patrón de Almacenamiento Cache	No		Top 2

Fuente: Elaboración propia

En la tabla 19 se elaboró una descripción y análisis de los patrones de diseño que se seleccionaron desde la perspectiva al construir la arquitectura de la aplicación.

En la tabla se muestra un listado con las publicaciones científicas más resaltantes referenciados en la presente investigación, de los cuales se describe el autor, el nombre del proyecto, los patrones de diseños utilizados, y una puntuación de rendimiento otorgado, el cual representa el nivel de rendimiento definido por el autor del proyecto, con una escala de 3, representada como:

Top 1: Patrón de diseño que presento un mejor rendimiento en la investigación.

Top 2: Patrón de diseño que presento el segundo mejor rendimiento en la investigación.

Top 3: Patrón de diseño que presento el rendimiento más bajo en la investigación.

También se clasifica asignando si es o no un patrón de diseño de arquitectura, representada como:

Si: Es Patrón de Diseño de Arquitectura.

No: No es Patrón de Diseño de Arquitectura.

Después de haber puntuado a los patrones de diseño de cada investigación se remarcó y se seleccionó los patrones de diseño de arquitectura debido a que son utilizadas para construir la arquitectura de la aplicación y los demás patrones de diseño no fueron seleccionados debido a que son utilizadas para desacoplar el software y no se pueden comparar entre sí o compararlos con los patrones de arquitectura.

A continuación, se muestra la lista de los patrones de diseño de arquitectura que ofrecieron un mejor rendimiento (Top 1) o los patrones de diseño que fueron agrupados entre sí para brindar un mejor rendimiento, los cuales fueron los siguientes:

1. Modelo, Vista y Controlador (MVC)
2. Modelo, Vista y Presentador (MVP)
3. Presentacion, Abstraccion y Control (PAC)
4. Modelo, Vista y VistaModelo (MVVM)
5. Precontrolador, Modelo, Vista, Controlador, Postcontrolador (MVC extendible)
6. MVC variádico

No solo los patrones de diseño pueden ser implementados solos, sino que además se pueden agrupar o combinar con otros patrones obteniendo un mejor rendimiento, así lo demuestra las investigaciones que se encontraron. Se mostrará a continuación, un cuadro de los patrones de diseño que son de arquitectura y el tiempo que demora al desarrollar una aplicación.

Tabla 26.

*Cantidad de Aplicaciones Web a desarrollar*

<b>Patrones de diseño de Arquitectura</b>	<b>Numero de aplicaciones</b>	<b>Estimación de tiempo al desarrollar</b>
<b>Modelo, Vista y Controlador (MVC)</b>	1	8 semanas
<b>Modelo, Vista y Presentador (MVP)</b>	1	8 semanas
<b>Presentacion, Abstraccion y Control (PAC)</b>	1	8 semanas
<b>Modelo, Vista y VistaModelo (MVVM)</b>	1	8 semanas
<b>Precontrolador, Modelo, Vista, Controlador, Postcontrolador (MVC extendible)</b>	1	8 semanas
<b>Modelo, Vista y Controlador Variádico (MVC Variádico)</b>	1	8 semanas

Fuente: Elaboración propia

La Tabla 20 muestra un listado sobre la estimación de tiempo que se desarrollara cada aplicación web con cada patrón de diseño de arquitectura, considerando que las aplicaciones no solo tendrán operaciones CRUP (crear, leer, actualizar y borrar) sino que además tendrá 4 procesos (venta directa, ventas por mesa, pedidos mediante la web y reservación web), lo cual la aplicación web será compleja y funcionable para ser implementado.

Se decidió elegir solo 2 patrones de diseño porque si se tomaran 4 patrones de diseño se obtendrían 4 aplicaciones con un estimado de tiempo de 40 semanas.

El tiempo de demora que llevara desarrollar las 4 aplicaciones no son factibles ya que la investigación debe concluir en el mes de agosto; es por eso que se decidió seleccionar solo 2 patrones de diseño con un tiempo de demora de 16 semanas, lo que hace posible concluir la investigación en el tiempo estimado.

Teniendo en cuenta por qué se eligieron desarrollar solo 2 aplicaciones web con cada patrón de diseño de arquitectura, se argumentará la elección de los 2 patrones de diseño a base las siguientes conclusiones:

Tabla 27.

*Selección de los Patrones de Diseño*

<b>Patrón de diseño de Arquitectura</b>	<b>Aplicabilidad</b>	<b>Criterios de rendimiento</b>	<b>Agrupamiento</b>	<b>Framework basados en el patrón de diseño</b>	<b>Framework web basados en el patrón de diseño</b>
<b>Modelo, Vista y Controlador (MVC)</b>	Sistema interactivo que los contenidos e interfaces sean dinámicos.	<ul style="list-style-type: none"> <li>• Buen orden en estructura de código.</li> <li>• Buen soporte al implementarlo.</li> <li>• Buen tiempo en realizar operaciones los usuarios.</li> <li>• Buen rendimiento de usabilidad.</li> </ul>	Buen rendimiento	Muchos como Laravel (PHP), Spring (Java), ASP.NET MVC (.NET), y más.	Variedad como Laravel, CodeIgniter, CakePHP, y más.
<b>Modelo, Vista y Presentador (MVP)</b>	Sistema interactivo que los contenidos e interfaces sean dinámicos.	<ul style="list-style-type: none"> <li>• Buena estructura de código.</li> <li>• Al agruparlo con otros patrones de diseño ofrece un buen tiempo en realizar operaciones los usuarios.</li> <li>• Es mantenible.</li> </ul>	Buen rendimiento	Muchos como Web de Google Toolkit, ZK, JFace, y más.	Web de Google Toolkit y más.

<b>Presentacion, Abstraccion y Control (PAC)</b>	Sistemas interactivos que realizan tareas múltiples.	<ul style="list-style-type: none"> <li>• Deficiencia en su estructura de su código.</li> <li>• Buen rendimiento al agruparlo con otros patrones de diseño.</li> </ul>	Buen rendimiento	-	-
<b>Modelo, Vista y VistaModelo (MVVM)</b>	Sistemas interactivos con la misma lógica de negocio pero con diferente demandas de UI.	<ul style="list-style-type: none"> <li>• Es mantenible.</li> <li>• Facilidad al desarrollarlo.</li> </ul>	-	Poco como MvvmCross (Android)	-
<b>Precontrolador, Modelo, Vista, Controlador, Postcontrolador (MVC extensible)</b>	Sistema interactivo que los contenidos e interfaces sean dinámicos.	<ul style="list-style-type: none"> <li>• Buen rendimiento en el tiempo que demora por solicitud al realizar operaciones.</li> </ul>	-	-	-
<b>Modelo, Vista y Controlador Variádico (MVC Variádico)</b>	Sistema interactivo que los contenidos e interfaces sean dinámicos.	<ul style="list-style-type: none"> <li>• Buen rendimiento al evitar redundancia entre sus componentes</li> </ul>	-	-	-

Fuente: Elaboración propia

La Tabla muestra un listado de los patrones de diseño de arquitectura, de los cuales se seleccionó 2 patrones de diseño:

Primero: no se seleccionaron los patrones de diseño modificados como son el Precontrolador, Modelo, Vista, Controlador, Postcontrolador (MVC extendible) y Modelo, Vista y Controlador Variádico (MVC Variádico) debido a que la base de su arquitectura es el patrón de diseño MVC (Modelo, Vista y Controlador) de tal manera que han sido modificados para obtener mejores resultados y al realizar una comparación con los otros patrones de diseño tienen la ventaja de que han sido mejorados o agregados otros componentes en su arquitectura.

Segundo: no se seleccionó el patrón de diseño Presentación, Abstracción y Control (PAC) debido a que este patrón no se está implementando últimamente en el desarrollo web.

Tercero: tenemos a 3 patrones de diseño a seleccionar que son Modelo, Vista y Controlador (MVC); Modelo, Vista y Presentador (MVP) y Modelo, Vista y VistaModelo (MVVM) de los cuales según el autor (Syromiatnikov & Weyns, 2014) están relacionados debido a que su diseño es Model View - \*(MV\*); de los cuales se seleccionó el patrón de diseño MVC Y MVP debido a que estos patrones han sido y siguen siendo implementados y utilizados en el desarrollo web como por ejemplo según el autor (Gu & Tang, 2010) se implementaron los dos patrones en el framework Webforms desarrollado por ASP.NET aplicándolo a la web; además existen más framework webs y se está desarrollando aplicaciones móviles que están basados en la arquitectura de dichos patrones de diseño. Por tal motivo y debido a que son patrones de diseño que son muy utilizados y documentados en el desarrollo web, se han seleccionado dichos patrones.

### **3.3.3. Desarrollar una aplicación utilizando los dos patrones de diseños**

#### **3.3.3.1. Selección del caso de estudio**

Existen diferentes casos de estudios para implementar los patrones de diseño.

Para lograr ello, se elaboró una descripción y análisis de los casos de estudio a seleccionar.

Tabla 28.

*Selección de Caso de Estudio*

<b>Aplicaciones web a desarrollar</b>	<b>Acceso a la información de la empresa existente</b>	<b>Realiza transacciones</b>	<b>Utilización de base de datos</b>	<b>Procesos de las aplicaciones web</b>	<b>Interacción cliente - administrador</b>	<b>Operaciones por día</b>
<b>Aplicaciones estáticas para diferentes empresas</b>	Si se tiene acceso de información y datos para realizar la aplicación.	<b>No</b>	<b>No</b> hace uso de base de datos.	No se selecciona como caso de estudio ya que no realiza ningún proceso y lo que realiza es mostrar al cliente la información.	<b>Si</b> se realiza interacción ya que el cliente solo lee texto informativo de productos o más.	<b>No</b> realiza procesos por día.
<b>Sistema de Ventas para ferretería</b>	Si se tiene acceso de información y datos para realizar la aplicación	<b>Si</b>	<b>Si</b> hace uso de base de datos.	<b>Si</b> se selecciona como caso de estudio ya que se realiza 1 proceso que es la venta directa.	<b>Si</b> se realiza interacción ya que el cliente solo interactúa de forma directa con el administrador y así registra la venta del cliente.	<b>Si</b> , diariamente el administrador realizara operaciones de forma directa con el cliente.

<b>Sistema de Ventas, Pedidos y Reservaciones a la Pollería</b>	Si se tiene acceso de información y datos para realizar la aplicación.	<b>Si</b>	<b>Si</b> hace uso de base de datos.	<b>Si</b> se selecciona como caso de estudio ya que realiza 4 procesos que es la venta directa, ventas por mesa, pedidos por la web y reservaciones de mesas por la web.	<b>Si</b> se realiza interacción ya que el cliente puede interactuar con el administrador de muchas maneras ya sea de forma directa (personalmente) o desde su móvil o computadora ya sea para realizar pedidos o reservaciones.	<b>Si</b> , diariamente las operaciones que realice el cliente se realizaran no solo de forma directa sino que también desde otro lugar y se podrá realizar una operación diferente para cada cliente.
<b>Sistema de Almacén para constructora</b>	No se tiene acceso de información y datos para realizar la aplicación.	<b>Si</b>	<b>Si</b> hace uso de base de datos.	<b>Si</b> se selecciona como caso de estudio ya que se realiza 1 proceso que es de recepcionar y almacenar los productos hasta el momento de su salida.	<b>Si</b> se realiza interacción ya que el administrador (comprador) registra los productos de los clientes (proveedores) y los almacena hasta su salida.	<b>No</b> , las operaciones se realizan periódicamente dependiente de los productos que no cuenten con stock en el almacén.

<b>Pagos por internet</b>	<b>No</b> se tiene acceso de información y datos para realizar la aplicación.	<b>Si</b>	<b>Si</b> hace uso de base de datos.	<b>Si</b> se selecciona como caso de estudio ya que se realiza 1 proceso que es el de realizar transferencias entre el cliente y el administrador mediante una entidad financiera autorizada por ambos.	<b>Si</b> se realiza interacción ya que el cliente (comprador) paga mediante una tarjeta y lo transfiere a la cuenta del administrador (vendedor).	<b>Si</b> , diariamente el cliente realiza pagos mediante una tarjeta.
<b>Sistema de matrícula y notas de una escuela</b>	<b>No</b> se tiene acceso de información y datos para realizar la aplicación.	<b>Si</b>	<b>Si</b> hace uso de base de datos.	<b>Si</b> , se realiza 2 procesos que es el de matricular y colocar sus respectivas notas a los alumnos.	<b>Si</b> se realiza interacción ya que el administrador solo interactúa de forma directa con el cliente de tal manera que registra y matricula al alumno.	<b>No</b> , las operaciones se realizan periódicamente ya sea para matricular o colocar notas.

Fuente: Elaboración propia

La Tabla 22 muestra un listado de diferentes aplicaciones a desarrollar como caso de estudio en la investigación, teniendo en cuenta los criterios evaluados para seleccionar el caso de estudio, se seleccionó el sistema propuesto para una pollería (Sistema de Ventas directa, Ventas por mesa, Pedidos vía web y Reservaciones vía web), en el que se tendrá 4 procesos lo que permite ser más complejo; a diferencia de las demás aplicaciones que tienen 1 o 2 procesos. Hay que tener en cuenta el acceso de información que nos tendremos de la empresa nos ayudara a avanzar la aplicación más rápida ya que nos facilitara los requerimientos funcionales para solucionar dicho problema.

Al implementar las aplicaciones se decidió seleccionar una problemática que exista en el mundo real, es por eso que brindara solución a la pollería "Richard", debido a que no cuenta con un control adecuado de ventas, pedidos y reservaciones, además nos brindara y facilitara la información que se requiera en cualquier momento; por lo que si seleccionamos realizar otro sistema se tendrá que buscar la empresa que necesite solucionar el mismo problema y que nos brinde el tiempo para obtener información que se requiera.

Una de las características que tendrá la aplicación propuesta es la interacción del cliente con la pollería el cual será de diferentes procesos ya sea de forma directa o indirecta.

Se desarrollará dos Aplicaciones Web de Ventas, Pedidos y Reservaciones a la Pollería, en la primera aplicación web se utilizará el patrón de diseño MVC y en la otra aplicación se usará el patrón de diseño MVP.

### **3.3.3.2. Selección de la metodología ágil de desarrollo de software**

Existen diferentes tipos de metodología ágil para utilizar e implementar en el desarrollo del software.

Para lograr ello, se elaboró en la tabla 4 una breve descripción en la que se analiza los diferentes tipos de metodologías ágiles existentes, de los cuales se seleccionó una metodología ágil.

Tabla 29.

*Selección de la metodología ágil de desarrollo de software*

<b>Metodología Ágil</b>	<b>Descripción</b>	<b>Programador / Relación con el Usuario</b>	<b>Etapas / Fases</b>	<b>Ventajas</b>	<b>Desventajas</b>	<b>Tipos de empresas</b>
<b>Extreme Programming (XP)</b>	Plantea un grupo de prácticas técnicas, ya que al aplicarlas sincronizadamente buscan resaltar los efectos positivos al desarrollar el software.	La comunicación se realiza mediante reuniones que son cara a cara, permitiendo encontrar soluciones en conjunto a los problemas que podamos recomendar.	Planeación Diseño Codificación Prueba	Entregas breves que permiten al cliente tener una mejor experiencia con el software. Al realizar rediseños que sugieren los clientes permiten mejorar internamente la estructura de un diseño sin tener que cambiar la funcionalidad.	Al rediseñar el software continuamente nos permite consumir mucho tiempo y recursos.  Al realizar entregas sobre el software, el cliente puede realizar cambios en el software por lo que puede modificarlo, y así conllevar más tiempo en volver a realizar los cambios y mostrarlos.	Pequeñas y medianas empresas.
<b>Desarrollo impulsado por las características (DIC)</b>	Está basada en la colaboración de los integrantes del equipo, la administración de complejidad de los problemas y los detalles técnicos que	La comunicación mediante reuniones en las que se entrega bloques pequeños de la funcionalidad del software.	Desarrollar un modelo general Elaborar una lista de características Plan según las características	Al realizar entregas de bloques pequeños de funcionalidad lo que le permite al usuario revisar mejor para buscar ambigüedades, errores u omisiones.	Falta de información en los diseños.	Grandes empresas

---

se comunican en forma verbal, grafico o en texto.

Diseño  
Construir

**SCRUM**

Integra un grupo de patrones de proceso poniendo énfasis en las prioridades del proyecto, en unidades de trabajo agrupada, la comunicación y la retroalimentación usual en el cliente.

Las reuniones se realizan diariamente con el cliente con un tiempo de 15 minutos por lo general.

Planeación  
  
Diseño  
  
Cierre del Proyecto

Al finalizar el sprint (unidades de trabajo), no es posible introducir cambios en el software. Entregas preliminares al cliente de tal manera que pueda demostrar el incremento y avance del software.

Demasiadas reuniones pueden conllevar al cansancio de los integrantes del proyecto.  
  
Si el cliente no brinda los requerimientos adecuados al problema, puede conducir a realizar mal el software.

Pequeñas y medianas empresas.

---

Fuente: Elaboración propia

La Tabla 23 muestra un listado sobre los diferentes tipos de metodologías ágiles, de los cuales se seleccionó la metodología ágil Extreme Programming (XP) ya que según (Lacovelli, 2008) es una de las metodologías más utilizadas en el desarrollo del software; una de las ventajas es que durante el proceso de pruebas y planificación, sus tasa de error es muy pequeña debido a que el cliente ha tenido la facilidad de entregas preliminares del software lo que le permite tener la facilidad de aceptar o modificar algún requerimiento que se hizo inicialmente; la comunicación que existe entre el cliente y los desarrolladores es continuamente por lo que el cliente tiene el control sobre las prioridades.

– **Fases de la Metodología Extreme Programming (XP)**

La metodología Extreme Programming (XP) está compuesto con las siguientes fases:

- Fase de Planeación
- Fase de Diseño
- Fase de Codificación
- Fase de Prueba

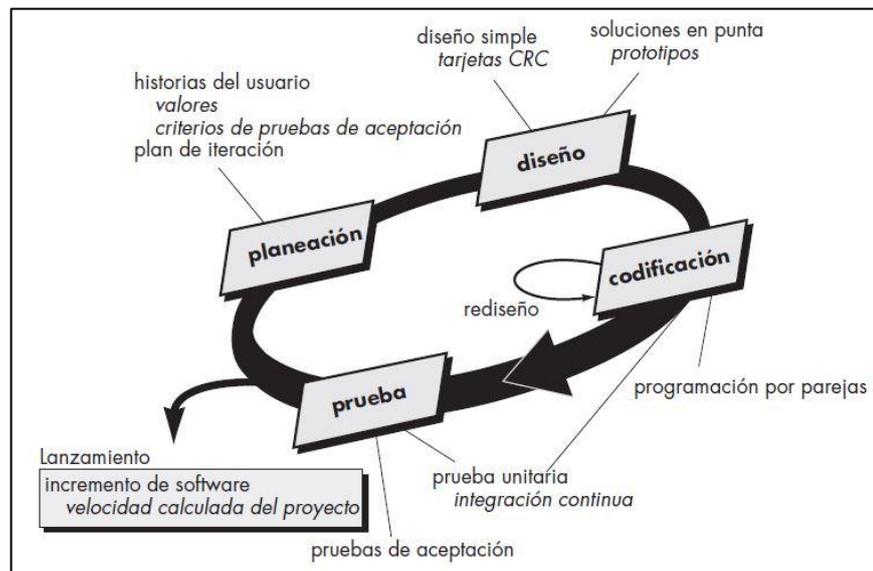


Figura 83: Fases de la Metodología Extreme Programming (XP).  
 Fuente: (Pressman, 2010)

### 3.3.3.3. Selección del lenguaje de programación

Existen diferentes tipos de lenguaje de programación para utilizar e implementar al desarrollar la aplicación web.

Para lograr ello, se elaboró en la tabla 24 una breve descripción en la que se analiza los diferentes tipos de lenguaje de programación existentes, de los cuales se seleccionó el lenguaje de programación a desarrollar en la aplicación.

Tabla 30.

*Selección del lenguaje de programación*

<b>Lenguaje de Programación</b>	<b>Paradigma</b>	<b>Descripción</b>	<b>Sistema Operativo</b>	<b>Licenciatura</b>
<b>PHP</b>	Se puede utilizar por programación orientada a objetos (POO), o también realizar una mezcla entre ambas.	Es un código abierto muy popular y conveniente para el desarrollo web y además se le puede incrustar HTML.	Multiplataforma: macOS, Linux, RISC OS, Unix, Microsoft Windows y probablemente muchos más.	Licencia PHP.
<b>Java</b>	Orientada a Objetos	Este lenguaje de programación nos permite escribir el software en una plataforma y luego ejecutarlo virtualmente en otra.	Multiplataforma: Linux, Unix, Microsoft Windows, RISC OS, macOS y más.	GNU GLP / Java Comunity Process.
<b>JavaScript</b>	Programación interpretada por lo que no se necesita compilarlos para ejecutarlos. Es decir, los programas que estén escritos en JavaScript se puede probar directamente en cualquier navegador.	Es utilizada principalmente para la creación de páginas dinámicas.	Probar directamente desde cualquier navegador.	GNU, Software libre.

<b>Python</b>	Su estructura de datos son eficientes y de un alto nivel conteniendo un enfoque simple pero es efectivo a la programación orientada a objetos.	Es poderoso y se aprende con facilidad. Su sintaxis es elegante y su tipado dinámico.	Multiplataforma: Linux, Unix, Microsoft Windows y macOS.	Licencia general de GNU, código abierto
<b>Ruby</b>	Programación funcional como la imperativa, orientada a objetos	Es un lenguaje de programación flexible, lo que permite que sus usuarios puedan alterarlos libremente.	Multiplataforma: Linux, Microsoft Windows y macOS.	Licencia libre
<b>Perl</b>	Programación orientada a objetos, procedimental y funcional.	Cuenta con herramientas potentes que sirven para procesar texto de tal manera que lo hace ideal al trabajar con HTML, XML y todos los demás lenguajes marcado y naturales.	Multiplataforma: Linux, Microsoft Windows y macOS.	Código abierto, Licencia Artística o GPL

Fuente: Elaboración propia

La Tabla 24 muestra un listado sobre los diferentes tipos de lenguajes de programación, de los cuales se seleccionó el lenguaje de programación PHP ya que es uno de los lenguajes más utilizadas en el desarrollo de software web; una de las ventajas es que su paradigma es orientado a objetos, además de ser software libre lo que me permite ser utilizado sin ninguna restricción como son las licencias. Además, el desarrollador tiene conocimientos al utilizar el lenguaje de programación PHP no permite avanzar más rápido el desarrollo del software.

#### **3.3.3.4. Desarrollo de la aplicación utilizando la metodología XP**

Esta aplicación se desarrollará en base a las fases del ciclo de vida de la Metodología Extreme Programming (XP), con el fin de describir el desarrollo del Sistema Web de Ventas, Pedidos y Reservaciones para la Pollería “Richard”. El uso de la metodología que emplearemos implica la realización de actividades por cada una de sus fases, a continuación, se describe por cada fase de la metodología, las actividades que se realizaran a lo largo de todo el desarrollo. Las fases de la metodología XP son las siguientes: Fase de Planificación, Fase de Diseño, Fase de Codificación y Fase de Prueba.

##### **a) Fase de Planificación**

###### **– Definición del problema**

La carencia de tecnología y automatización en ciertos procesos para tener el control adecuado en Ventas, Pedidos y Reservaciones de la pollería, han generado muchos inconvenientes al adquirir informes como las ventas, pedidos y reservaciones que se realizan diariamente o mensualmente.

- La falta de información de los clientes en cuanto a los pedidos que hace continuamente.
- La falta de automatización de los procesos dificultó el adecuado a la información.
- No tener un control adecuado de ventas en la pollería dificultó su respectivo registro.
- Al no controlar adecuadamente las reservaciones en la pollería dificultó su respectivo registro.
- La falta de un control adecuado de recepción y pedidos por mesa.

– **Visión del proyecto**

Desarrollar una aplicación web para la Pollería “Richard”, que me permita tener un control adecuado de todas sus Ventas, Pedidos y Reservaciones.

Proveer información detallada de los productos que se han vendido.

Generar informes que nos brinden información de los productos, ventas, pedidos y reservaciones que se realicen en la pollería.

– **Roles**

Los miembros de un equipo trabajan mejor cuando hay roles establecidos, cada rol tiene responsabilidades. En nuestro proyecto, se asignaron roles al equipo de desarrollo.

Tabla 31.

*Roles del Equipo del Proyecto*

---

<b>ROLES DEL EQUIPO DEL PROYECTO</b>	
<b>Programador:</b>	Christian Erick Gonzales Gonzales
<b>Clientes:</b>	Antonio Gonzales Ballena
<b>Tester</b>	Christian Erick Gonzales Gonzales
<b>(Probadores):</b>	
<b>Trackear</b>	
<b>(Responsable del seguimiento):</b>	Christian Erick Gonzales Gonzales

---

Fuente: Elaboración propia

– **Levantamiento de Requerimientos**

La información se obtuvo por una ficha de requerimientos que fue realizado a la Pollería “Richard” de la ciudad de Monsefú. Requerimientos Funcionales iniciales.

- La aplicación web se adapta cualquier dispositivo.
- La aplicación web contiene una interfaz gráfica agradable para el usuario.
- La aplicación web permite gestionar a los usuarios y así brindar privilegios a cada uno.
- La aplicación web gestiona el módulo de usuarios, productos, categoría de productos, mesas, contactos, venta directa, ventas por mesa, reservaciones y pedidos online.

- En el módulo de ventas se gestiona la información de las ventas directas que se realicen diariamente cuando el cliente realice su pedido y pague por su pedido, y se retira de la pollería.
- En el módulo de ventas por mesa se gestiona la información de las ventas por mesa al realizar el cliente cada pedido en la mesa correspondiente, realiza el pago y se retira de la mesa.
- En el módulo de pedidos por la web se gestiona la información de cada cliente que realiza pedidos de los productos que desean, los pedidos se harán mediante la web de la pollería.
- En el módulo de reservación se gestiona la información de los clientes que han reservado una mesa en la pollería, las reservaciones se harán mediante la web de la pollería.
- La aplicación web permite registrar, modificar y eliminar usuarios.
- Se gestiona y se detalla la información en una tabla dinámica para cada subproceso.

– **Historias de usuario**

Es la base del éxito del producto, en el cual se muestra mediante un listado de características la necesidad del cliente y encontrarlos en el producto final. Esta investigación está distribuida en iteraciones y en cada una de ellas hay una entrega.

En estas historias detallas las actividades a realizar en cada una de ellas, y así obtener un plan de entregas, y en cada entrega se desarrollará partes de la aplicación web completamente funcionales.

Después de las reuniones con el cliente y definirse los requerimientos en el sistema y sus niveles de acceso, se describen las siguientes historias de usuario.

Tabla 32.

*Historia de Usuario 1: Página de bienvenida y Acceso al sistema*

<b>Historia de Usuario</b>	
<b>Numero:</b> 01	<b>Nombre de Historia de Usuario:</b> Página de bienvenida y Acceso al sistema
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>Antes de haber iniciado la aplicación web, se mostrará los diferentes servicios que ofrece la pollería y además mostrará un acceso al usuario, el cual debe contar con una cuenta previamente creada, al acceder a la aplicación se mostrará una pantalla de acceso donde se requerirá su usuario y su contraseña, y así validar los datos en la aplicación.</p> <p>Si no ingresa al sistema es porque después de varios intentos, deberá comunicarse con el administrador para que pueda acceder nuevamente.</p> <p><b>Observaciones:</b></p> <p>Para acceder a la aplicación, se debe comunicar con el administrador, de tal manera que se le crea una cuenta de acuerdo a su puesto de trabajo.</p> <p>La estructura de la aplicación se ha realizado pensando en el crecimiento de la pollería.</p>	

Fuente: Elaboración propia

Tabla 33.

*Historia de Usuario 2: Usuarios y validación de usuarios*

Historia de Usuario	
<b>Numero:</b> 02	<b>Nombre de Historia de Usuario:</b> Usuarios y validación de usuarios
<b>Usuario:</b> Todos	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>El usuario podrá acceder a los diferentes módulos que el administrador le ha brindado al crear su usuario y además de cerrar sesión.</p> <p>Si el usuario no tiene cuenta no podrá acceder a la aplicación, lo cual le mostrará un mensaje de “ingresar nuevamente”.</p> <p><b>Observaciones:</b></p> <p>En la barra superior (menú), se encuentra un icono que permitirá al usuario salir de la aplicación.</p>	

Fuente: Elaboración propia

Tabla 34.

*Historia de Usuario 3: Permisos de Usuario*

Historia de Usuario	
<b>Numero:</b> 03	<b>Nombre de Historia de Usuario:</b> Permisos de Usuario
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>El usuario tendrá distintos permisos y privilegios, el cual permitirá tener un mejor control del sistema.</p> <p>Los tipos de usuario son:</p> <p>El tipo de usuario administrador, cuenta con todos los privilegios creados para el control del sistema.</p> <p>El tipo de usuario cajero, cuenta con el control de ventas, ventas por mesas, pedidos online y clientes.</p> <p>El tipo de usuario mozo, cuenta con el control de mesas, reservaciones, productos, ventas por mesa, ventas y clientes.</p>	

Fuente: Elaboración propia

Tabla 35.

*Historia de Usuario 4: Registro y lista de usuarios*

<b>Historia de Usuario</b>	
<b>Numero:</b> 04	<b>Nombre de Historia de Usuario:</b> Registro y lista de usuarios
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>El administrador al crear el usuario en un formulario completara los datos que son: nombre de usuario, grupo de usuario, nombre completo, apellido completo, email, imagen o foto, contraseña, confirmación de la contraseña y estado (activo o inactivo).</p> <p>Además de registrar, se podrá modificar, eliminar y visualizar en una tabla el nombre del usuario, estado (activo o inactivo) y la fecha de registro del usuario creado.</p> <p><b>Observaciones:</b></p> <p>Solo el administrador podrá crear, modificar o eliminar al usuario.</p>	

Fuente: Elaboración propia

Tabla 36.

*Historia de Usuario 5: Sección de clientes*

<b>Historia de Usuario</b>	
<b>Numero:</b> 05	<b>Nombre de Historia de Usuario:</b> Sección de clientes
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrá visualizar mediante una tabla los nombres y apellidos del cliente, DNI, teléfono y tipo de cliente (eventual o fijo); además de realizar búsquedas para encontrar al cliente específico.	

Fuente: Elaboración propia

Tabla 37.

*Historia de Usuario 6: Registro de Clientes*

<b>Historia de Usuario</b>	
<b>Numero:</b> 06	<b>Nombre de Historia de Usuario:</b> Registro de clientes
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrá registrar y visualizar a los clientes nuevos. Al registrar un nuevo cliente se mostrará un formulario para que complete los datos que son: nombre y apellidos del cliente, DNI, teléfono y tipo de cliente (eventual o fijo). Además de registrar, se podrá modificar y eliminar.	

Fuente: Elaboración propia

Tabla 38.

*Historia de Usuario 7: Sección de Productos*

<b>Historia de Usuario</b>	
<b>Numero:</b> 07	<b>Nombre de Historia de Usuario:</b> Sección de productos
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>Se podrá visualizar mediante una tabla el nombre, precio del producto y categoría de producto (bebidas, pollos a la brasa o platos) y la imagen del producto; además de realizar búsquedas para encontrar el producto específico.</p>	

Fuente: Elaboración propia

Tabla 39:

*Historia de Usuario 8: Registro de Productos*

<b>Historia de Usuario</b>	
<b>Numero:</b> 08	<b>Nombre de Historia de Usuario:</b> Registro de productos
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>Se podrán registrar y visualizar los productos nuevos. Al registrar un nuevo producto se mostrará un formulario para que complete los datos que son: nombre, precio del producto, categoría de producto (bebidas, pollos a la brasa o platos) y la imagen del producto. Además de registrar, se podrá modificar y eliminar.</p>	

Fuente: Elaboración propia

Tabla 40.

*Historia de Usuario 9: Sección de Categorías de Productos*

<b>Historia de Usuario</b>	
<b>Numero:</b> 09	<b>Nombre de Historia de Usuario:</b> Sección de categoría de productos
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrá visualizar mediante una tabla el nombre de las categorías de productos.	

Fuente: Elaboración propia

Tabla 41.

*Historia de Usuario 10: Registro de Categorías de Productos*

<b>Historia de Usuario</b>	
<b>Numero:</b> 10	<b>Nombre de Historia de Usuario:</b> Registro de categoría de productos
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b>                      Al registrar una nueva categoría del producto se mostrará un formulario para que complete el dato nombre de la categoría.                      Además de registrar, se podrá modificar y eliminar.</p> <p><b>Observaciones:</b>                      El administrador no brindara privilegios a nadie para el registro de categoría de productos.</p>	

Fuente: Elaboración propia

Tabla 42.

*Historia de Usuario 11: Sección de Mesas*

Historia de Usuario	
<b>Numero:</b> 11	<b>Nombre de Historia de Usuario:</b> Sección de mesas
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrá visualizar mediante una tabla el número de mesa, descripción, tipo de mesa (familiar, vip, pareja, normal), cantidad de sillas y estado (disponible u ocupado).	

Fuente: Elaboración propia

Tabla 43.

*Historia de Usuario 12: Registro de mesas*

Historia de Usuario	
<b>Numero:</b> 12	<b>Nombre de Historia de Usuario:</b> Registro de mesas
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrán visualizar y registrar las mesas. Al registrar una nueva mesa se muestra el formulario para que complete los datos que son: número de mesa, descripción, tipo de mesa (familiar, vip, pareja, normal) y cantidad de sillas. Además de registrar, se podrá modificar y eliminar.	
<b>Observaciones:</b> El administrador brindara privilegios al mozo para el registro de mesas.	

Fuente: Elaboración propia

Tabla 44.

*Historia de Usuario 13: Sección de reservaciones de mesa vía web*

Historia de Usuario	
<b>Numero:</b> 13	<b>Nombre de Historia de Usuario:</b> Sección de reservaciones de mesa vía web
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrá visualizar mediante una tabla la fecha de la reservación, hora de la reservación, código de verificación, cliente, cantidad de sillas, teléfono y observaciones.	

Fuente: Elaboración propia

Tabla 45.

*Historia*

*de Usuario 14: Registro de reservaciones de mesa vía web*

Historia de Usuario	
<b>Numero:</b> 14	<b>Nombre de Historia de Usuario:</b> Registro de reservaciones de mesa vía web
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> El administrador y el mozo podrán registrar y visualizar las reservaciones de las mesas vía web. Al registrar una reservación de mesa vía web se mostrará un formulario para que complete los datos que son: fecha de la reservación, hora de la reservación, cantidad de personas, nombre y apellidos del cliente, DNI y teléfono. Además de registrar, se podrá eliminar.	
<b>Observaciones:</b> El administrador brindara privilegios al mozo para el registro de reservaciones vía web.	

Fuente: Elaboración propia

Tabla 46.

*Historia de Usuario 15: Sección de ventas directa y ventas por mesa*

Historia de Usuario	
<b>Numero:</b> 15	<b>Nombre de Historia de Usuario:</b> Sección de ventas directa y ventas por mesa
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> Se podrá visualizar mediante una tabla el código de la venta, fecha de la venta, comprobante (factura o boleta), cliente, mesa de la venta, sub total, igv, total, vendedor, estado de la venta (facturada)	

Fuente: Elaboración propia

Tabla 47.

*Historia de Usuario 16: Registro de las ventas directas*

Historia de Usuario	
<b>Numero:</b> 16	<b>Nombre de Historia de Usuario:</b> Registro de las ventas directas
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<b>Descripción:</b> El administrador, el mozo y el cajero podrán registrar y visualizar las ventas directas. Al registrar venta directa se mostrará un formulario para que complete los datos que son: código de la facturación, fecha de venta, comprobante (boleta o factura), nombre del cliente, producto, precio y cantidad de producto.	
<b>Observaciones:</b> El administrador brindara privilegios al mozo y al cajero para el registro de ventas directa.	

Fuente: Elaboración propia

Tabla 48.

de Usuario 17: Registro de las ventas por mesa

Historia de Usuario	
<b>Numero:</b> 17	<b>Nombre de Historia de Usuario:</b> Registro de las ventas por mesa
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b> El administrador, el mozo y el cajero podrán registrar y visualizar las ventas por mesa. Al registrar venta por mesa se mostrará todas las mesas disponibles, al realizar la venta se mostrará un formulario para que complete los datos que son: código de la facturación, numero de la mesa seleccionada, fecha de venta, comprobante (boleta o factura), nombre del cliente, producto, precio y cantidad de producto.</p> <p><b>Observaciones:</b> El administrador brindara privilegios al mozo y al cajero para el registro de ventas por mesa.</p>	

Fuente: Elaboración propia

Tabla 49.

Historia de Usuario 18: Sección de pedidos online

Historia de Usuario	
<b>Numero:</b> 18	<b>Nombre de Historia de Usuario:</b> Sección de pedidos online
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 5
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b> Se podrá visualizar mediante una tabla la fecha del pedido, código de verificación del pedido, nombre del cliente, DNI del cliente, teléfono, nombre del producto y cantidad del producto.</p>	

Fuente: Elaboración propia

Tabla 50:

*Historia de Usuario 19: Registro de Pedidos Online*

Historia de Usuario	
<b>Numero:</b> 19	<b>Nombre de Historia de Usuario:</b> Registro de pedidos online
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 5
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b> El administrador y el cajero podrán registrar y visualizar los pedidos online Al registrar pedidos online se mostrará todos los productos disponibles, al realizar el pedido se mostrará un formulario para que complete los datos que son: cantidad del producto seleccionado, nombre del cliente, DNI del cliente y teléfono.</p> <p><b>Observaciones:</b> El administrador brindara privilegios al cajero para la visualización de los pedidos online.</p>	

Fuente: Elaboración propia

Tabla 51.

*Historia de Usuario 20: Sección de Contactar*

Historia de Usuario	
<b>Numero:</b> 20	<b>Nombre de Historia de Usuario:</b> Sección de contactar
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 6
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b> Se podrá visualizar mediante una tabla el nombre del cliente, correo y comentario sobre alguna experiencia en la pollería.</p>	

Fuente: Elaboración propia

Tabla 52:

*Historia de Usuario 21: Registro de Contactar*

Historia de Usuario	
<b>Numero:</b> 21	<b>Nombre de Historia de Usuario:</b> Registro de contactar
<b>Usuario:</b> Antonio Gonzales Ballena	<b>Iteración Asignada:</b> 6
<b>Prioridad en Negocio:</b> Alta	<b>Riesgos en Desarrollo:</b> Media
<b>Programador responsable:</b> Christian Erick Gonzales Gonzales	
<p><b>Descripción:</b></p> <p>Los clientes podrán enviar comentarios o sugerencias sobre la pollería en la opción contactar que se encuentra en la página principal de la pollería, se mostrará un formulario para que complete los datos que son: nombre del cliente, email y comentario.</p>	

Fuente: Elaboración propia

– **Planificación de los lanzamientos**

Después de realizar las historias de los usuarios, la planificación del lanzamiento se hace como resultado de esto. Después de evaluar los riesgos, las prioridades y realizar estimaciones basadas en el pico, se establece el plan en colaboración con el cliente.

Tabla 53.

*Planificación de los lanzamientos*

Nº	Historias de Usuario	Iteración					
		1	2	3	4	5	6
1	Página de bienvenida y Acceso al sistema	X					
2	Usuarios y validación de usuarios	X					
3	Permisos de Usuario		X				
4	Registro y lista de usuarios		X				
5	Sección de clientes		X				
6	Registro de clientes		X				
7	Sección de productos		X				
8	Registro de productos		X				
9	Sección de categoría de productos		X				
10	Registro de categoría de productos		X				
11	Sección de mesas		X				
12	Registro de mesas		X				
13	Sección de reservaciones de mesa vía web			X			
14	Registro de reservaciones de mesa vía web			X			
15	Sección de ventas directa y ventas por mesa				X		
16	Registro de las ventas directas				X		

17	Registro de las ventas por mesa				X		
18	Sección de pedidos online					X	
19	Registro de pedidos online					X	
20	Sección de contactar						X
21	Registro de contactar						X

Fuente: Elaboración propia

– **Iteraciones**

En la tabla se muestra el plan de iteraciones que se preparó con el cliente, estableciendo las iteraciones en ocho semanas que se estimaron para la finalización del proyecto.

Tabla 54.

Iteraciones del Patrón de Diseño MVC

Iteraciones del MVC	Nº Historia de Usuario	Historias de Usuario	Semanas								
			1	2	3	4	5	6	7	8	
1	01	Página de bienvenida y Acceso al sistema									
	02	Usuarios y validación de usuarios									
2	03	Permisos de Usuario									
	04	Registro y lista de usuarios									

	05	Sección de clientes								
	06	Registro de clientes								
	07	Sección de productos								
	08	Registro de productos								
	09	Sección de categoría de productos								
	10	Registro de categoría de productos								
	11	Sección de mesas								
	12	Registro de mesas								
3	13	Sección de reservaciones de mesa vía web								
	14	Registro de reservaciones de mesa vía web								
4	15	Sección de ventas directa y ventas por mesa								
	16	Registro de las ventas directas								
	17	Registro de las ventas por mesa								
5	18	Sección de pedidos online								
	19	Registro de pedidos online								
6	20	Sección de contactar								
	21	Registro de contactar								

Fuente: Elaboración propia

Tabla 55.

Iteraciones del Patrón de Diseño MVP

Iteraciones del MVP	Nº Historia de Usuario	Historias de Usuario	Semanas								
			9	10	11	12	13	14	15	16	
1	01	Página de bienvenida y Acceso al sistema									
	02	Usuarios y validación de usuarios									
2	03	Permisos de Usuario									
	04	Registro y lista de usuarios									
	05	Sección de clientes									
	06	Registro de clientes									
	07	Sección de productos									
	08	Registro de productos									
	09	Sección de categoría de productos									
	10	Registro de categoría de productos									
	11	Sección de mesas									
	12	Registro de mesas									
3	13	Sección de reservaciones de mesa vía web									
	14	Registro de reservaciones de mesa vía web									
4	15	Sección de ventas directa y ventas por mesa									

	16	Registro de las ventas directas								
	17	Registro de las ventas por mesa								
5	18	Sección de pedidos online								
	19	Registro de pedidos online								
6	20	Sección de contactar								
	21	Registro de contactar								

Fuente: Elaboración propia

Las historias de usuarios se agruparon en iteraciones de acuerdo con su complejidad, es decir, un desarrollo incremental basado en la implementación de los requisitos iniciales del proyecto.

– **Architectural Spike**

Las herramientas que se utilizaron en el desarrollo de la aplicación son gratuitas por lo que me permiten crear una aplicación web.

- Xampp: En su versión 5.6.32, su plataforma es un servidor web, que cuenta con licenciamiento gratuito. Esta basado principalmente en el sistema gestión de base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script, php y Perl.

- Administradores de Base de Datos: MySQL Workbench y PhpMyAdmin, son herramientas que se utilizan para el manejo de la administración de la Base de Datos y el modelado de esta.
- Editores de código: Sublime Text, Atom y Brackets, estos soportan una gran variedad de lenguajes de programación.

## **b) Fase de Diseño**

En el proyecto se pusieron en prácticas las consideraciones de diseño que están propuestos por la metodología.

### **– Modelamiento de la base de datos**

Teniendo en cuenta los requerimientos funcionales de la aplicación, es importante tener un modelo que nos permita cumplir con los requisitos establecidos.

- **Modelo Entidad Relación**

Mediante este modelo podemos representar simplificadaamente los componentes que intervienen en un proceso de negocio y el modo que están relacionados entre sí. Este es un modelo inicial implementado en el sistema de gestión de base de datos.

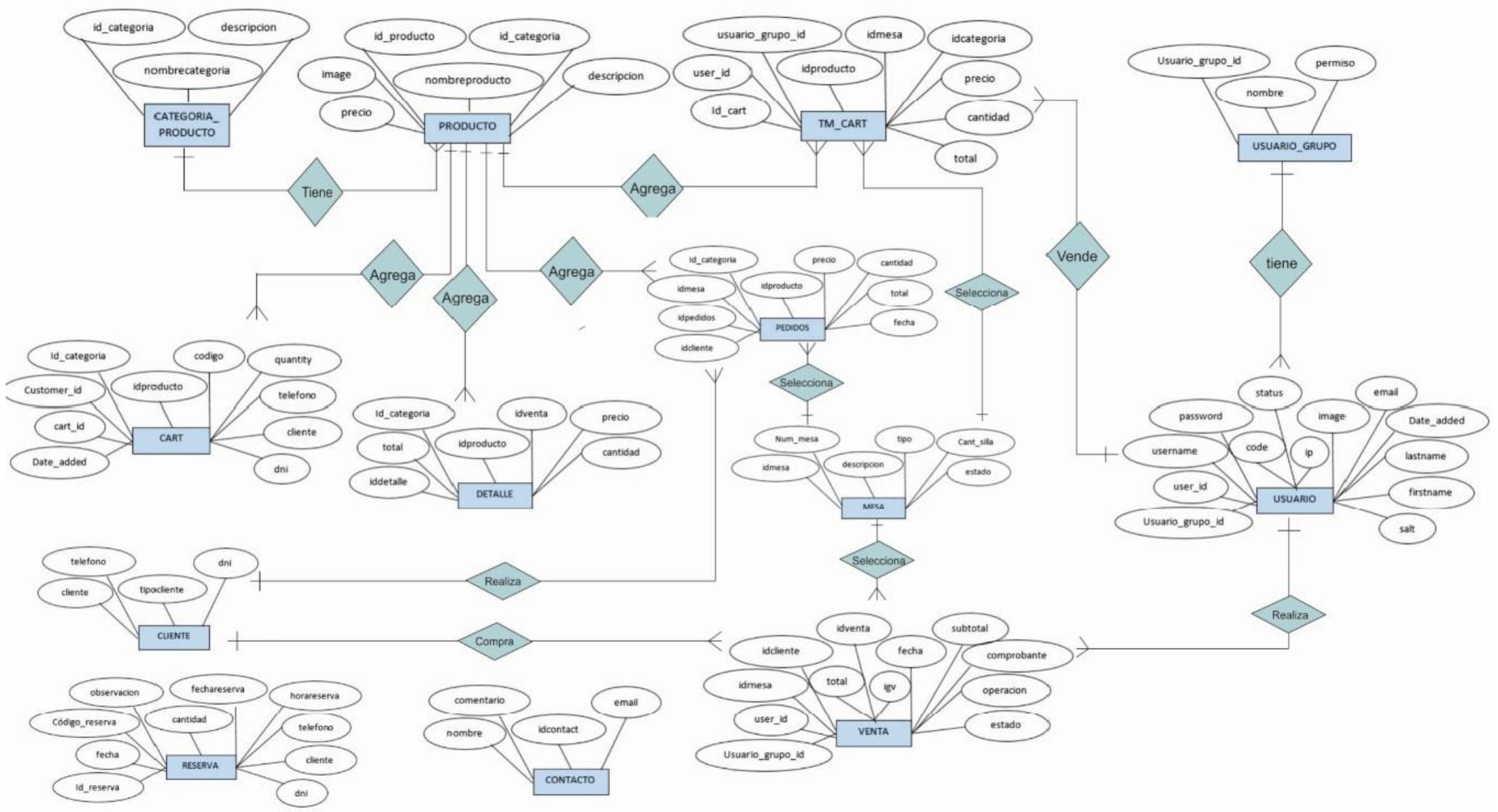


Figura 84. Modelo Entidad Relación. Fuente: Elaboración propia





– **Interfaces de la aplicación web**

La interfaz de la aplicación web está relacionada con las iteraciones de las historias de usuario.

- Página de bienvenida y Acceso al sistema



Figura 87. Interfaz de Página de Inicio. Fuente: Elaboración propia.

En la página de bienvenida se muestra los diferentes servicios que brinda la pollería, no solo se realiza las ventas en el local de la pollería, sino que además se puede realizar pedidos y reservaciones de mesas.

Ingrese los datos de su cuenta

Usuario

Contraseña

Ingresar

Figura 88. Acceso al Sistema. Fuente: Elaboración propia

Al acceder al sistema de la pollería, si el usuario tiene cuenta accederá al sistema rápidamente, pero sino puede ingresar el usuario se le mostrará un mensaje de advertencia de ingreso no valido.

- Interfaz de la sección clientes

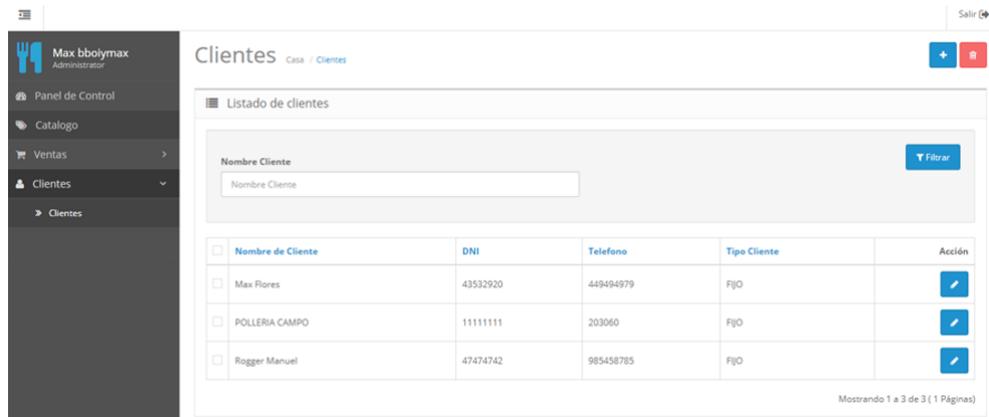


Figura 89. Interfaz de la sección Clientes. Fuente: Elaboración propia

Mediante una tabla se mostrará a todos los clientes registrados en el sistema, además de realizar búsquedas a un cliente específico.

- Registro de los clientes

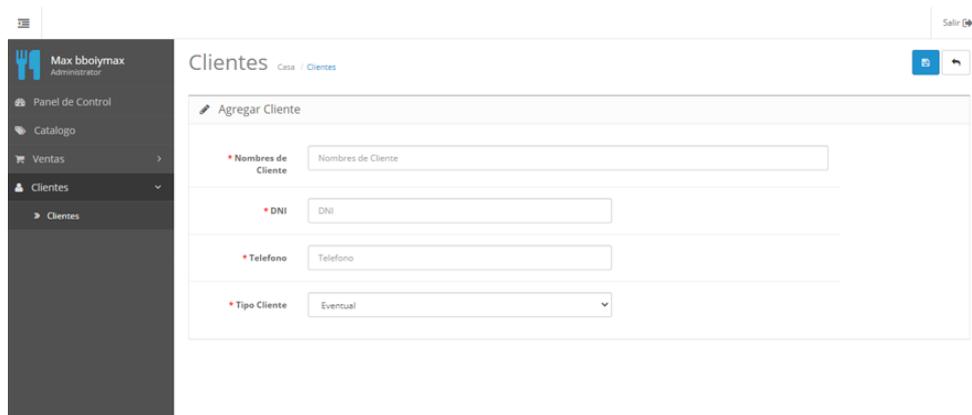


Figura 90. Registro de los Clientes. Fuente: Elaboración propia

Se podrá agregar, editar y eliminar los usuarios; se registran datos del cliente, así como el tipo de cliente (eventual o fijo).

- Interfaz de la sección de productos

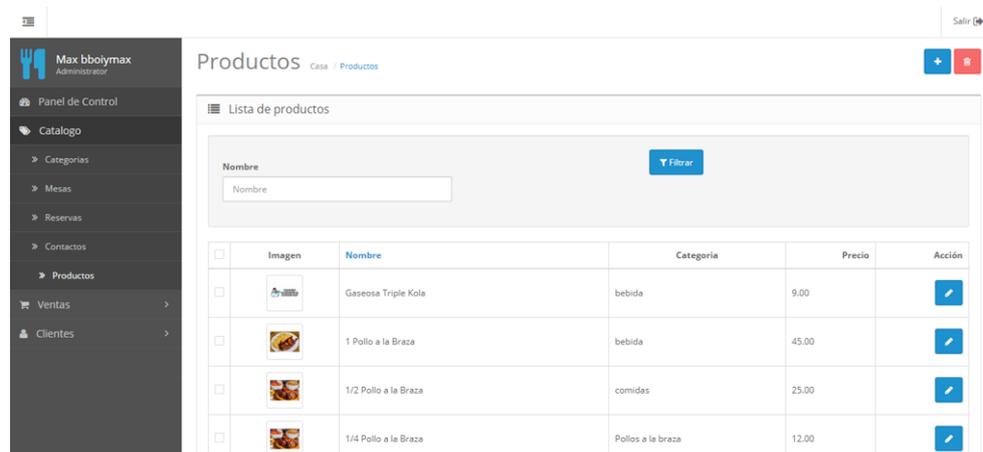


Figura 91. Interfaz de la sección de productos. Fuente: Elaboración propia

Mediante una tabla se mostrará a todos los productos registrados en el sistema, además de realizar búsquedas a un producto específico.

- Registro de los productos

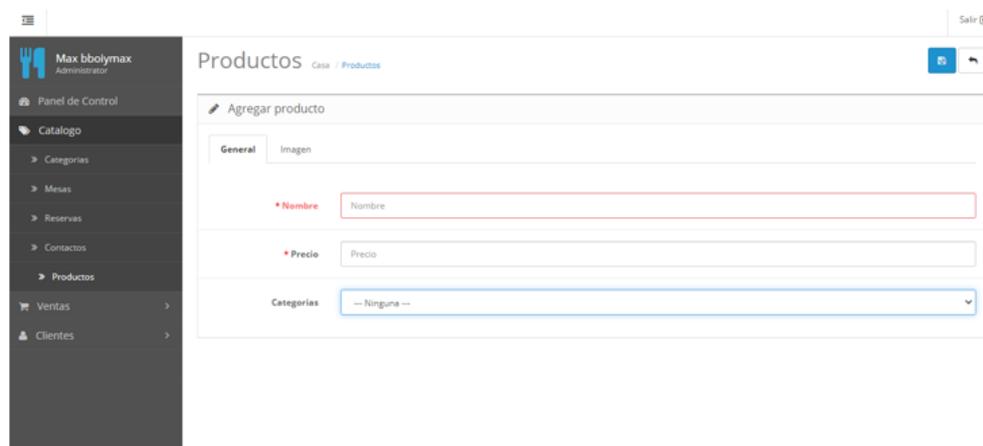


Figura 92. Registro de los productos. Fuente: Elaboración propia

Se podrá agregar, editar y eliminar los productos; se registran datos del producto, así como la imagen del producto y a la categoría a la que pertenece (bebidas, pollos a la brasa, platos).

- Interfaz de la sección de categoría de productos

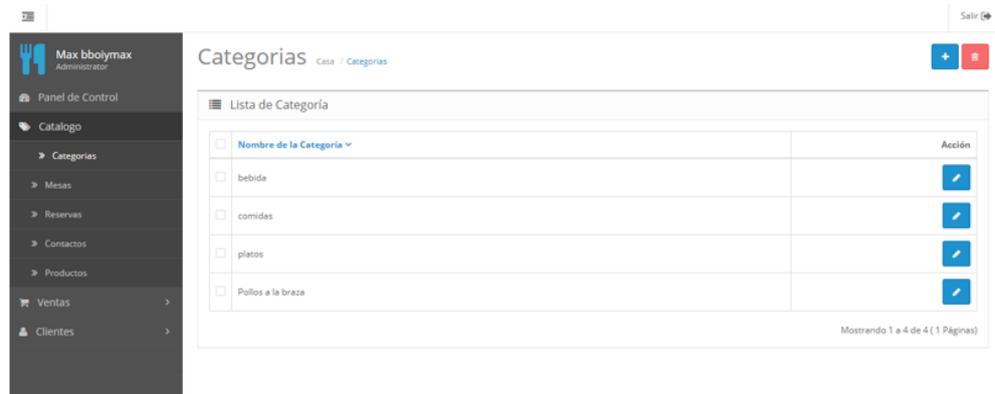


Figura 93. Interfaz de la sección de categoría de productos. Fuente: Elaboración propia

Mediante una tabla se mostrará a todas categorías de los productos registrados en el sistema.

- Registro de la categoría de productos

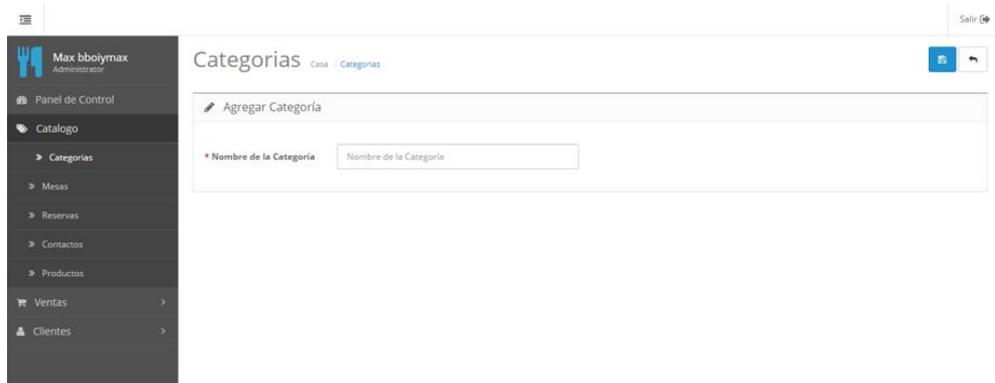
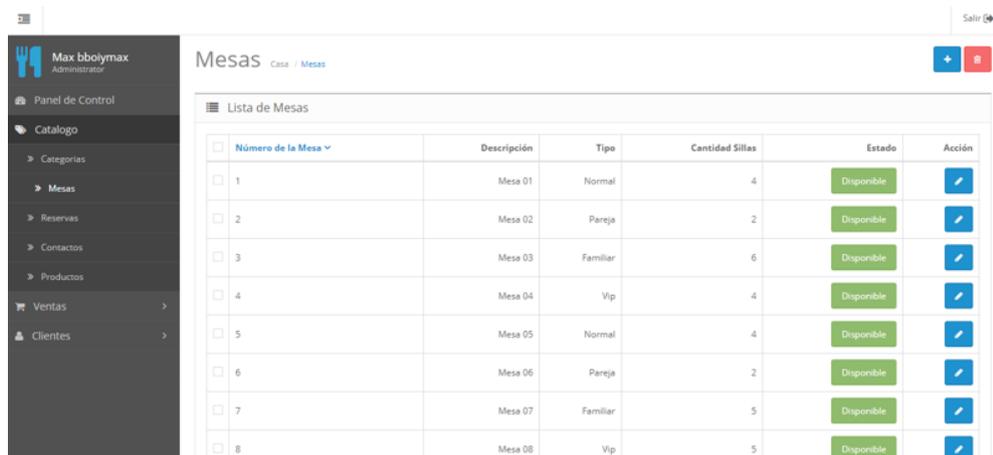


Figura 94. Registro de la categoría de productos. Fuente: Elaboración propia

Se podrá agregar, editar y eliminar los productos; se registra el nombre de la categoría de producto.

- Interfaz de la sección de mesas



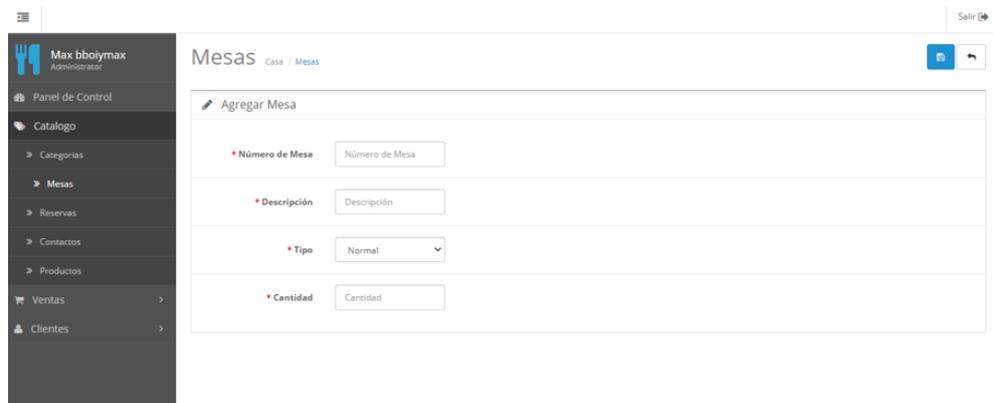
The screenshot shows the 'Mesas' section of the Max bbolymax Administrator interface. It features a sidebar menu on the left with options like 'Panel de Control', 'Catalogo', 'Mesas', 'Reservas', 'Contactos', 'Productos', 'Ventas', and 'Clientes'. The main content area displays a table titled 'Lista de Mesas' with the following data:

Número de la Mesa	Descripción	Tipo	Cantidad Sillas	Estado	Acción
1	Mesa 01	Normal	4	Disponible	[Edit]
2	Mesa 02	Pareja	2	Disponible	[Edit]
3	Mesa 03	Familiar	6	Disponible	[Edit]
4	Mesa 04	Vip	4	Disponible	[Edit]
5	Mesa 05	Normal	4	Disponible	[Edit]
6	Mesa 06	Pareja	2	Disponible	[Edit]
7	Mesa 07	Familiar	5	Disponible	[Edit]
8	Mesa 08	Vip	5	Disponible	[Edit]

Figura 95. Interfaz de la sección de mesas. Fuente: Elaboración propia

Mediante una tabla se mostrará a todas mesas registrados en el sistema.

- Registro de las mesas



The screenshot shows the 'Agregar Mesa' form in the 'Mesas' section. The form includes the following fields:

- Número de Mesa:** Input field with placeholder 'Número de Mesa'.
- Descripción:** Input field with placeholder 'Descripción'.
- Tipo:** Dropdown menu with 'Normal' selected.
- Cantidad:** Input field with placeholder 'Cantidad'.

Figura 96. Registro de las Mesas. Fuente: Elaboración propia

Se podrá agregar, editar y eliminar las mesas; se registran datos de la mesa, así como el tipo de mesa a la que pertenece (normal, pareja, vip, familiar).

- Interfaz de reservaciones de mesas vía web

RICHARD

INICIO BEBIDA COMIDAS PLATOS POLLOS A LA BRAZA RESERVAS CONTACTAR

## RESERVACIONES

completa el siguiente formulario para solicitar una reservación

Fecha de Reserva: Fecha de Reserva

\* Hora de Reserva: Hora 00:00

Cantidad de Personas: 1

\* Nombre y Apellidos: Nombre

\* DNI: DNI

\* Teléfono: Teléfono

RESERVAR AHORA

Figura 97. Interfaz de reservaciones de mesas vía web. Fuente: Elaboración propia

En la página de bienvenida de la pollería, se encuentra la opción de “RESERVAS”, los usuarios se podrán registrar para reservar una mesa; al completar los datos aparecerá un código de reservación el cual será presentado en la pollería para acceder a la reserva.

- Visualización y Registro de reservaciones de mesas vía web

Fecha	Hora	Hora	Cliente	Cantidad	Telefono	Observaciones	Acción
2021-02-12	15:30	RES-uy6m0azchi	Christian Górciales	5	985647130	VIA WEB	[Icono]
2021-01-21	15:30	RES-ct7oivgrm	Luis Guevara	5	985647130	VIA WEB	[Icono]
2020-11-30	10:50	RES-ko6ygaabv	max flores	1	747454484	VIA WEB	[Icono]

Figura 98. Visualización y Registro de reservaciones de mesas vía web.  
Fuente: Elaboración propia

Se podrá agregar una reserva mediante la web y se muestra en una tabla todas las reservas que se han hecho en el día.

- Interfaz de la sección de ventas directas y ventas por mesa

ID VENTA	FECHA	COMPROBANTE	CLIENTE	MESA	SUBTOTAL	IGV	TOTAL	VENDEDOR	ESTADO
1	2020-11-30	BOLETA	Max Flores	1	10.00	0.00	10.00	admin	Facturada
2	2020-12-19	BOLETA	POLLERIA CAMPO	3	0.00	0.00	0.00	admin	Facturada
3	2020-12-19	BOLETA	Max Flores	3	0.00	0.00	0.00	admin	Facturada
4	2020-12-19	BOLETA	Rogger Manuel	3	0.00	0.00	0.00	admin	Facturada
5	2021-01-08	BOLETA	POLLERIA CAMPO	1	25.00	0.00	25.00	admin	Facturada
6	2021-01-20	BOLETA	Max Flores	1	53.00	0.00	53.00	admin	Facturada
7	2021-01-20	BOLETA	Max Flores	1	90.00	0.00	90.00	admin	Facturada
8	2021-01-20	BOLETA	Rogger Manuel	2	50.00	0.00	50.00	admin	Facturada
9	2021-02-01	BOLETA	POLLERIA CAMPO	1	12.00	0.00	12.00	admin	Facturada
10	2021-02-03	BOLETA	Max Flores	1	450.00	0.00	450.00	admin	Facturada
11	2021-02-03	BOLETA	POLLERIA CAMPO	1	0.00	0.00	0.00	admin	Facturada
12	2021-02-03	BOLETA	Max Flores	1	0.00	0.00	0.00	admin	Facturada

Figura 99. Interfaz de la sección de ventas directas y ventas por mesa.  
Fuente: Elaboración propia

Mediante una tabla se mostrará todas las ventas registradas en el sistema ya sea venta directa o venta que se realizó por cada mesa.

- Registro de las ventas directas



Figura 100. Registro de las ventas directas. Fuente: Elaboración propia

Se podrá agregar una venta; se registran datos como el cliente, tipo de comprobante (boleto o factura) y los productos que desea adquirir con sus respectivos precios y cantidad.

- Registro de las ventas por mesa

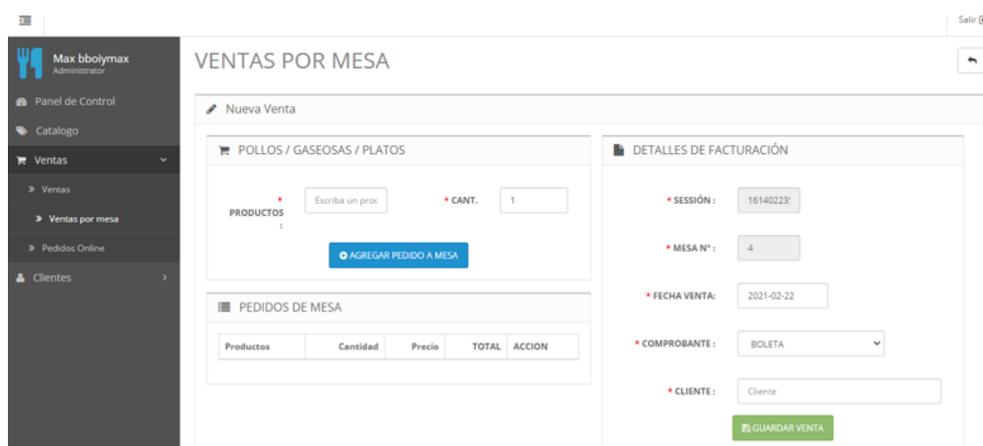


Figura 101. Registro de las ventas por mesa. Fuente: Elaboración propia

Se podrá agregar una venta por mesa; se selecciona la mesa y se registran datos como el cliente, tipo de comprobante (boleta o factura) y los productos que desea adquirir con sus respectivos precios y cantidad.

- Interfaz de pedidos online



Figura 102. Interfaz de pedidos online. Fuente: Elaboración propia



Figura 103. Interfaz del pedido de un producto. Fuente: Elaboración propia

En la página de bienvenida se mostrará variedad de platos o bebidas que ofrece la pollería, al seleccionar un plato se muestra un formulario; al completar los datos aparecerá un código de pedido el cual será presentado en la pollería para acceder al pedido.

- Interfaz de los registrados de pedidos online

The screenshot shows the 'Pedidos Online' interface. On the left is a dark sidebar with navigation options: 'Max bboymax Administrador', 'Panel de Control', 'Catalogo', 'Ventas' (with sub-options: 'Ventas', 'Ventas por mesa', 'Pedidos Online'), and 'Clientes'. The main content area is titled 'Pedidos Online' and contains a table of online orders.

Fecha	Código	Cliente	DNI	Teléfono	Producto	Cant
2020-11-25 23:20:04	k6rnfz7y1v9i	Max	43532920	945398547	1/2 Pollo a la Braza	1
2020-11-26 22:05:47	yjgmuuzie0ss	max flores	45985478	945874585	1 Pollo a la Braza	1
2020-11-28 14:06:52	7w36d28cuyrg	rogger	85478545	985878587	1/2 Pollo a la Braza	1
2020-11-28 15:07:58	g4wz7sand26x	ss	22323232	945439974	1/2 Pollo a la Braza	1
2020-11-29 22:37:41	5xzdln1ddxh4	wwdwdwdwdwd	43532920	985147845	1/2 Pollo a la Braza	1
2021-01-20 09:34:34	r1hyen020gs	Carlos Chumioque	45224240	971321032	1 Pollo a la Braza	1
2021-01-20 09:41:55	1f3rdpk1p2rp	Jimmy Gonzales	44008566	971321032	1 Pollo a la Braza	2
2021-01-20 09:42:38	dgrdthg1bpfr	Carlos Chumioque	42540690	4646455	1/8 Pollo a la Braza	5
2021-01-20 09:43:02	8ym5j5kb4gwi	Christian G	45224240	971321032	1 Pollo a la Braza	4

Mostrando 1 a 9 de 9 ( 1 Páginas)

Figura 104. Interfaz de los registrados de pedidos online. Fuente: Elaboración propia

Mediante una tabla se mostrará todos los pedidos online que se han hecho en el día con los datos respectivos.

- Interfaz de contactar

The screenshot shows the 'Contactar' interface. On the left is a dark sidebar with navigation options: 'Max bboymax Administrador', 'Panel de Control', 'Catalogo', 'Categorias', 'Mesas', 'Reservas', 'Contactos', 'Productos', 'Ventas', and 'Clientes'. The main content area is titled 'Contactar' and contains a table of contact records.

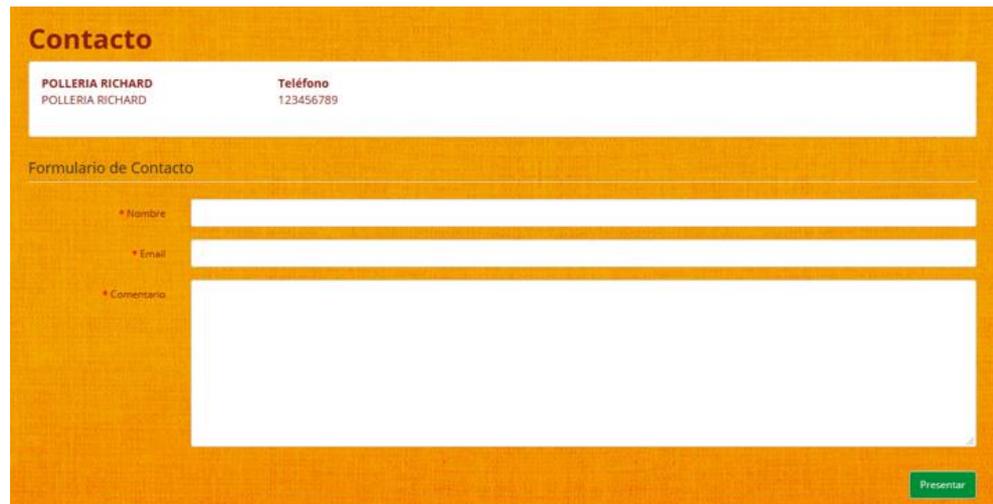
Nombre	Correo	Comentario
Christian Erik	crg_0410@hotmail.com	vghgfhghf
Jimmy Gonzales	tucorreo@tudominio.com	mas comentarios
promoveranoseguro	bboymax3@hotmail.com	jhhjh jbojnhnk
wdp*/8z/09<script>	max@interberica.com	wdp*/8z/09<script>

Mostrando 1 a 4 de 4 ( 1 Páginas)

Figura 105. Interfaz de contactar. Fuente: Elaboración propia

Mediante una tabla se mostrará a todos los comentarios o sugerencias que han enviado los usuarios a la pollería.

- Interfaz de contactar



The image shows a contact form titled "Contacto" for "POLLERIA RICHARD". At the top, there is a white box containing the business name "POLLERIA RICHARD" and the phone number "Teléfono 123456789". Below this is the "Formulario de Contacto" section, which includes three input fields: "Nombre", "Email", and "Comentario". Each field has a red asterisk indicating it is required. A green "Presentar" button is located at the bottom right of the form.

Figura 106. Interfaz de contactar. Fuente: Elaboración propia

Se podrá enviar mediante un formulario comentario o sugerencias que tienen los usuarios hacia la pollería.

#### d) Fase de Prueba

Las pruebas son requerimientos para cumplir con los objetivos de las historias de usuario, estas nos sirven para saber si se ha implementado lo que en realidad se deseaba. Este proceso está dirigida al funcionamiento de cada iteración que fue definida.

Tabla 56.

*Prueba de aceptación: Pagina de bienvenida y Acceso al sistema*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 1	<b>Numero de historia 1,</b> Página de bienvenida y Acceso al sistema
<b>Nombre de la prueba:</b> Acceso al sistema con datos correctos	
<b>Descripción:</b> Se muestra una pantalla de bienvenida mostrando los diferentes servicios que brinda la pollería, para acceder al sistema web el usuario debe escribir su respectiva contraseña que se le asigno, se valida al usuario y se ingresa al sistema.	
<b>Condiciones de ejecución:</b> Conexión a internet, servidor web, base de datos activa y usuario registrado.	
<b>Entrada:</b> El usuario ingresa su usuario y contraseña que se fue asignada. La actividad muestra un cuadro de dialogo validando los datos y así mismo redireccionar a su perfil de trabajo.	
<b>Resultado esperado:</b> El servidor devuelve información según la información del usuario.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 57.

*Prueba de aceptación: interfaz y registro de clientes, interfaz y registro de productos, interfaz y registro de categoría de productos, interfaz y registro de mesas.*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 2	<b>Numero de historia 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12,</b> interfaz y registro de clientes, interfaz y registro de productos, interfaz y registro de categoría de productos, interfaz y registro de mesas.
<b>Nombre de la prueba:</b> Registrar, actualizar o eliminar datos del proceso.	
<b>Descripción:</b> Al estar activo la sesión del usuario puede visualizar, registrar, actualizar y eliminar datos en base a privilegios definidos por el administrador.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<p><b>Entrada:</b>                      El usuario puede hacer búsqueda de información en tiempo real con ayuda de tablas dinámicas (datatables).                      Se realiza el registro llenando el formulario, se actualiza los datos en las tablas dinámicas y se elimina los datos. Todas estas operaciones si el usuario cuenta con privilegios.                      Al ejecutar cualquiera de estas 3 operaciones, se muestra un cuadro emergente notificando el registro, actualización o eliminación de los datos. Dichas operaciones son visualizados en las tablas dinámicas (datatables).</p>	
<p><b>Resultado esperado:</b>                      La distribución de los datos en las tablas dinámicas (datatables) con sus respectivas operaciones.                      Registro, actualización y eliminación de los datos de las secciones.                      Datos actualizados en la Base de datos.</p>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 58.

*Prueba de aceptación: Sección y Registro de reservaciones de mesa vía web*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 3	<b>Numero de historia 13 y 14,</b> Sección y Registro de reservaciones de mesa vía web
<b>Nombre de la prueba:</b> Visualizar estructura de la sección de reservaciones de mesa vía web	
<b>Descripción:</b> La sección de reservaciones se encuentra activa para gestionar la información del proceso de reservaciones.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<p><b>Entrada:</b></p> <p>El usuario visualiza la interfaz gráfica de la sección de reservaciones. El cliente puede acceder a la aplicación web para registrar su reservación, lo realiza llenando el formulario para poder realizar reservación por internet. También se podrá realizar la reservación mediante una llamada telefónica a la pollería.</p> <p>Al ejecutar la operación de registro, se muestra un código aleatorio avisando que el registro fue correcto. El código aleatorio de confirmación será mostrado cuando se esté ubicado en la pollería y así luego asignarle su mesa correspondiente.</p> <p>Todos los datos están relacionados por un ID del proceso día de reservaciones. Mediante una tabla se muestran todos los registros de las reservaciones que se han realizado en todo el día.</p>	
<p><b>Resultado esperado:</b></p> <p>Sección de reservaciones realiza todo el proceso de las reservaciones, así como también las reservaciones telefónicas.</p> <p>Datos actualizados en la Base de datos.</p> <p>Los códigos de reservación tienen que ser iguales para acceder a la reservación de la mesa.</p>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 59.

*Prueba de aceptación: Sección de ventas directa y ventas por mesa*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 4	<b>Numero de historia 15,</b> Sección de ventas directa y ventas por mesa
<b>Nombre de la prueba:</b> Visualizar estructura de la sección de ventas directa y ventas por mesa	
<b>Descripción:</b> La sección de ventas directa y ventas por mesa está activo para gestionar la información del proceso de ventas en general.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<p><b>Entrada:</b>                      El usuario visualiza la interfaz gráfica de la sección de ventas.                      Al registrar una venta directa o venta por mesa, todas las ventas serán almacenadas en una sola tabla dinámica(datatables).                      El usuario visualiza la información en tiempo real con ayuda de tablas dinámicas (datatables).</p>	
<p><b>Resultado esperado:</b>                      Módulo de ventas realiza todo el proceso de las ventas directas y ventas por mesa.                      Datos actualizados en la Base de datos.</p>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 60.

*Prueba de aceptación: Registro de las ventas directas*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 5	<b>Numero de historia 16,</b> Registro de las ventas directas
<b>Nombre de la prueba:</b> Visualizar estructura del registro de las ventas directas	
<b>Descripción:</b> La sección de ventas directa está activo para gestionar la información del proceso de ventas directamente.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<p><b>Entrada:</b>                      El usuario visualiza la interfaz gráfica de la sección de ventas.                      El usuario que tiene privilegios para acceder a la sección de ventas, lo realiza llenando el formulario para poder realizar la venta directamente.                      Al referirnos de venta directa, el cliente llega a la pollería y realiza sus pedidos posteriormente sin necesidad de ocupar una mesa.                      Al ejecutar la operación de registro, se pedirán los datos del cliente y si no se encuentra en el sistema se registrará posteriormente el cliente.                      Todos los datos están relacionados por un ID del proceso de ventas.                      Mediante una tabla se muestran todos los registros de las ventas que se han realizado en todo el día.</p>	
<p><b>Resultado esperado:</b>                      Módulo de ventas realiza todo el proceso de las ventas directas.                      Datos actualizados en la Base de datos.</p>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 61.

*Prueba de aceptación: Registro de las ventas por mesa*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 6	<b>Numero de historia 17,</b> Registro de las ventas por mesa
<b>Nombre de la prueba:</b> Visualizar estructura del registro de las ventas por mesa	
<b>Descripción:</b> La sección de ventas por mesa se encuentra activa para la gestión de información del proceso de ventas por mesa.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<b>Entrada:</b> El usuario visualiza la interfaz gráfica de la sección de ventas. El usuario que tiene privilegios para acceder a la sección de ventas, lo realiza llenando el formulario para poder realizar la venta por mesa. Al referirnos de venta por mesa, el cliente llega a la pollería y ocupa una mesa disponible, después realiza los pedidos correspondientes al mozo. Al ejecutar la operación de registro, se pedirán los datos del cliente y si no se encuentra en el sistema se registrará posteriormente el cliente. Todos los datos están relacionados por un ID del proceso de ventas. Mediante una tabla se muestran todos los registros de las ventas que se han realizado en todo el día.	
<b>Resultado esperado:</b> Módulo de ventas realiza todo el proceso de las ventas por mesa. Datos actualizados en la Base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 62.

*Prueba de aceptación: Sección y Registro de pedidos online*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 7	<b>Numero de historia 18 y 19,</b> Sección y Registro de pedidos online
<b>Nombre de la prueba:</b> Visualizar estructura de la sección de pedidos online	
<b>Descripción:</b> La sección de pedidos online está activo para la gestión de información del proceso de pedidos.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<p><b>Entrada:</b>                      El usuario visualiza la interfaz gráfica de la sección de pedidos online.                      El cliente puede acceder a la aplicación web para registrar su pedido, lo realiza seleccionando el producto que desea adquirir y llenando el formulario para poder realizar pedido correspondiente.                      Al ejecutar la operación de registro, se muestra un código aleatorio avisando que el registro fue correcto. El código aleatorio de confirmación será mostrado cuando se esté ubicado en la pollería y así luego recibir su pedido correspondiente.                      Todos los datos están relacionados por un ID del proceso día de pedidos online. Mediante una tabla se muestran todos los registros de las pedido online que se han realizado en todo el día.</p>	
<p><b>Resultado esperado:</b>                      Datos actualizados en la Base de datos.                      Los códigos de pedidos online tienen que ser iguales para acceder al pedido correspondiente.</p>	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 63.

*Prueba de aceptación: Sección y Registro de contactar*

<b>PRUEBA DE ACEPTACIÓN</b>	
<b>Numero:</b> 8	<b>Numero de historia 20 y 21,</b> Sección y Registro de contactar
<b>Nombre de la prueba:</b> Visualizar estructura de la sección de contactar	
<b>Descripción:</b> La sección de contactar se encuentra activo para la información del proceso de contactar.	
<b>Condiciones de ejecución:</b> Sesión activa en el sistema, conexión a internet, base de datos activa.	
<p><b>Entrada:</b>                      El usuario visualiza la interfaz gráfica de la sección de contactar.                      El cliente puede acceder a la aplicación web para registrar algún reclamo, sugerencias o experiencia que ha pasado en la pollería, lo realiza llenando el formulario.                      Al ejecutar la operación de registro, se muestra un mensaje avisando que el registro fue correcto.                      Todos los datos están relacionados por un ID del proceso que identifica a los usuarios que han realizado su registro.                      Mediante una tabla se muestran todos los registros los usuarios que han enviado sus sugerencias o reclamos.</p>	
<b>Resultado esperado:</b> Datos actualizados en la Base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria.	

Fuente: Elaboración propia

## IV. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

Se concluyó que los indicadores de tiempo de desarrollo, líneas de código escritas, uso de memoria RAM, uso de CPU y tiempo de respuesta hacen referencia al diseño para desarrollar el software, debido a que si no tienes un buen diseño puede generar problemas de productividad con el paso del tiempo en tu software.

En este sentido, tras realizar la presente investigación se concluyó que el patrón de diseño MVC permite reducir significativamente tiempo de desarrollo, puesto que se obtuvo una mejor estimación en tiempo de desarrollo, líneas de código escritas, uso de memoria RAM, uso de CPU y tiempo de respuesta, con lo cual se rechaza la hipótesis planteada en esta investigación.

- a) Tras la revisión exhaustiva de la literatura se elaboró una tabla que contiene una lista de 6 patrones de diseño con una breve descripción, autor y año del cual se tomó como referencia para la documentación de cada patrón de diseño.
- b) Para establecer los patrones de diseños seleccionados, se utilizó un proceso de valoración utilizando los papers descritos en la investigación, seguido de un listado con los patrones de diseño utilizados y los que mejor rendimiento ofrecieron en sus respectivas investigaciones, quedando Modelo, Vista y Controlador (MVC) y Modelo, Vista y Presentador (MVP).
- c) Se implementaron dos pequeños proyectos de desarrollo de software: Desarrollo del Sistema Web para la gestión de Ventas y Pedidos de la pollería Richard del distrito de Monsefú, en la provincia de Chiclayo. Se llevaron a cabo la gestión de Venta Directa, Venta por Mesa, Pedidos Web y Reservaciones

Web, en el cual todos estos procesos están implementados con cada patrón de diseño.

- d) Se determinó el desempeño en el indicador de tiempo de desarrollo que tomo realizar en la fase de codificación del patrón de diseño MVC empleo 280 horas para generar los 21 requerimientos, mientras que en el patrón de diseño MVP empleo 320 horas para generar los mismos requerimientos generados en todo el proceso del proyecto. Así mismo en el indicador de líneas de código escritas que tomo realizar en la fase de codificación del patrón de diseño MVC empleo 3 098 líneas de código escritas para generar los requerimientos, mientras que en el patrón de diseño MVP empleo 7 587 líneas de código escritas para generar los mismos requerimientos del proyecto de software desarrollado.
  
- e) Se evaluó el uso de uso de Memoria RAM, se tiene que el patrón de diseño MVC utiliza un promedio 654 MB (megabit) del total de Memoria RAM que tiene 1024 MB de Memoria RAM la computadora en el cual se realizó las pruebas, mientras que en el patrón de diseño MVP utiliza un promedio de 598 MB de Memoria RAM. Además, en el indicador de uso de CPU, se tiene que el patrón de diseño MVC utiliza un promedio del 4% del total de uso de CPU que tiene la computadora en el cual se realizó las pruebas mediante procesos, mientras que en el patrón de diseño MVP utiliza un promedio de 5% de Uso de CPU. Así mismo En el indicador de Tiempo de Respuesta, se tiene que el patrón de diseño MVC al realizar una petición se obtuvo un promedio de tiempo de respuesta de 160,28 milisegundos por proceso, mientras que en el patrón de diseño MVP obtuvo un promedio de tiempo de respuesta de 253,86 milisegundos.

Estos indicadores nos permiten tener una mejor referencia al desarrollar un software, en este caso en el indicador de tiempo de desarrollo y líneas de código se determinó que el patrón de diseño MVP requiere mucho tiempo e implementación de más líneas de código lo que nos genera problemas de productividad en tiempo y dinero.

Al evaluar los indicadores de uso de memoria RAM, uso de CPU y tiempo de respuesta concluimos que al ejecutar diferentes procesos que realiza la aplicación el patrón de diseño MVP utiliza más cantidad de memoria RAM y uso de CPU permitiendo que la computadora se vuelva más lento al ejecutar los procesos; además el tiempo de respuesta para mostrar lo que solicita el usuario demora más que el patrón de diseño MVC

En este sentido, tras realizar la presente investigación se concluyó que el patrón de diseño MVC permite reducir significativamente tiempo de desarrollo, puesto que se obtuvo una mejor estimación en tiempo de desarrollo, líneas de código escritas, uso de memoria RAM, uso de CPU y tiempo de respuesta.

## **4.2 Recomendaciones**

La importancia de tener un buen diseño en el software, nos permite hacer uso de patrones de diseño obteniendo ciertas ventajas como el tener una mejor usabilidad, y ante la posibilidad de realizar un mantenimiento de software se hace más rápido y eficiente.

Estos patrones de diseño son una gran alternativa para los desarrolladores de software, pero la demasiada información en sitios webs, libros y más nos llevan a la confusión de realizar una selección adecuada, y en muchos casos es necesario combinar diferentes patrones para obtener un mejor resultado.

## REFERENCIAS

- [1] I. I. M. K. Chowdhury, "Pre and Post Controller based MVC Architecture for Web Application," *Informatics, Electron. Vis. (ICIEV), 2016 5th Int. Conf.*, pp. 552–557, 2016.
- [2] N. Sheng and Y. Wang, "The Software Design of Modular Instrument Based on Proxy Pattern," *Proc. 2018 Chinese Autom. Congr. CAC 2018*, no. 98, pp. 976–979, 2019.
- [3] P. Ouyang, W. Cao, M. Wu, C. Gan, and F. Wang, "Design of intelligent drilling system software framework and data architecture based on MVC pattern," *Chinese Control Conf. CCC*, vol. 2019-July, pp. 7075–7078, 2019.
- [4] Y. Elshater, P. Martin, and E. Hassanein, "Using Design Patterns to Improve Web Service Performance," *2015 IEEE Int. Conf. Serv. Comput.*, pp. 746–749, 2015.
- [5] R. Sharma and K. Davuluri, "Design patterns for Machine Learning Applications," *Proc. 3rd Int. Conf. Comput. Methodol. Commun. ICCMC 2019*, no. Iccmc, pp. 818–821, 2019.
- [6] L. Yingda, L. Jianzhuang, and C. Xiaowen, "Variadic Parameter Command Pattern and the Utilization in MVC," *Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS*, vol. 2018-November, pp. 374–377, 2019.
- [7] S. Caballe, J. A. Ortega, J. M. Camps, L. Barolli, E. Kulla, and E. Spaho, "A presentation framework to simplify the development of Java EE application thin clients," *Proc. - 2014 8th Int. Conf. Complex, Intell. Softw. Intensive Syst. CISIS 2014*, pp. 421–426, 2014.
- [8] G. . Guizzo, T. E. . b E. Colanzi, and S. R. . R. Vergilio, "A Feasibility Analysis for the Application of Design Patterns in Search Based Product Line Design," *2013 32nd Int. Conf. Chil. Comput. Sci. Soc.*, pp. 116–120, 2013.
- [9] A. Katzmaier and M. Hanneghan, "Design Pattern Evaluation of Mobile and Web Based Application Frameworks," *2013 Sixth Int. Conf. Dev. eSystems Eng.*, pp. 157–162, 2013.
- [10] E. Kazan, M. Canturk, and M. Bastan, "Performance analysis of a software developed with and without design patterns: A case study," *2015 Twelve Int. Conf. Electron. Comput. Comput.*, pp. 1–5, 2015.
- [11] X. Li, D. Chang, H. Pen, X. Zhang, Y. Liu, and Y. Yao, "Application of MVVM design pattern in MES," *2015 IEEE Int. Conf. Cyber Technol. Autom. Control Intell. Syst. IEEE-CYBER 2015*, no. 2012, pp. 1374–1378, 2015.

- [12] F. Honrubia, "Programación de aplicaciones para Iphone y Ipad," *Program. Apl. Iphone y Ipad*, p. 191, 2014.
- [13] I. Aedo, P. Díaz, S. Montero, and T. Zarranadía, "Patrones de diseño aplicados al desarrollo de Objetos Digitales Educativos (ODE)," *Patrones diseño Apl. al Desarro. Objetos Digit. Educ.*, p. 122, 2011.
- [14] J. Gamma, Erick; Helm, Richard; Johnson, Ralph; Vlissides, "Patrones de Diseño Elementos de software orientado a objetos reutilizable," *Patrones Diseño Elem. Softw. orientado a objetos Reutil.*, p. 1001, 1994.
- [15] G. Prabowo, H. Suryotrisongko, and A. Tjahyanto, "Una historia de dos enfoques de desarrollo: estudio empírico sobre la capacidad de mantenimiento y la modularidad de la aplicación móvil Android con patrón de diseño Anti-Pattern y Model-View-Presenter "A Tale of Two Development Approach: Empirical Study," *Proc. - 2nd 2018 Int. Conf. Electr. Eng. Informatics, ICELTICs 2018*, pp. 149–154, 2018.
- [16] D. G. Y. Fernández Romero Yenisleidy, "Patrón Modelo-Vista-Controlador," *Rev. Telem@tica*, vol. 11, no. 1, p. 11, 2012.
- [17] P. L. Thung, C. J. Ng, S. J. Thung, S. Sulaiman, and A. C. Study, "Improving a web application using design patterns: A case study," *Inf. Technol. (ITSim), 2010 Int. Symp.*, vol. 1, pp. 1–6, 2010.
- [18] Diéguez Fernando Rodríguez, "Integración de componentes software en páginas web," Enero, 2014.
- [19] Pérez Carvajal Rafael Jesús, "Mantenimiento del software (UF1894)," 2014.
- [20] C.G. Dictino, G.P. Sebastián Rubén and L.M. Emilio, "Introducción a la Informática básica," Enero, 2017.
- [21] Gómez Ruedas Jesús, "Dirección y gestión de proyectos de tecnologías de la información en la empresa," Enero, 2016.
- [22] Cabot Sagrera Jordi, "Ingeniería del software," 2013.
- [23] Cardador Cabello Antonio Luis, "Implantación de aplicaciones web en entornos internet, intranet y extranet (MF0493\_3)," Julio, 2014.
- [24] Ferrer Martínez Juan, "Implantación de aplicaciones Web," Enero, 2014.

- [25] M.P. Juan Carlos and R.P. Arturo Francisco, "Administración de Software de un Sistema Informático," Enero, 2014.
- [26] Eslava Muñoz Vicente Javier, "El nuevo php," Enero, 2018.
- [27] Robledo Clodoaldo and Robledo David, "Google Web Toolkit," Enero, 2013.
- [28] Hernandez Sampieri, Fernandez Collado and Baptista Lucio, 2010.

## ANEXOS

### ANEXO Nº 1 – RESOLUCIÓN DE AMPLIACIÓN DEL PROYECTO DE INVESTIGACIÓN



FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO  
RESOLUCIÓN N°0956-2021/FIAU-USS

Pimentel, 15 de octubre de 2021

**VISTO:**

El oficio N°0334-2021/FIAU-IS-USS de fecha 25 de septiembre de 2021, de la Dirección de Escuela profesional de INGENIERÍA DE SISTEMAS con el que remite el Acta de reunión N°2209-2021 del Comité de investigación de la referida Escuela profesional, acerca de la Tesis presentada por egresados del Programa de estudios de INGENIERÍA DE SISTEMAS, y;

**CONSIDERANDO:**

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N°2209-2021, de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y AMBIENTE, a cargo de egresados del Programa de estudios INGENIERÍA DE SISTEMAS, hasta el 22 de septiembre de 2022.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

**SE RESUELVE:**

**ARTÍCULO ÚNICO: AMPLIAR VIGENCIA**, de la Tesis a cargo de los egresados del Programa de estudios de **INGENIERÍA DE SISTEMAS** que se detallan en el anexo de la presente Resolución, hasta el 22 de septiembre de 2022.

**REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE**



Cc: Interesado, Archivo

## ANEXO Nº 1 – CARTA DE ACEPTACIÓN



### CARTA DE ACEPTACIÓN DE LA EMPRESA

Chiclayo 01 de abril del 2019

Mg. Ernesto Dante Rodríguez Lafitte  
Decano de la Facultad de Ingeniería, Arquitectura y Urbanismo  
Universidad Señor de Sipán

Presente.-

REF: Facultad de Ingeniería, Arquitectura y Urbanismo de la Universidad Señor de Sipán

Tengo el agrado de dirigirme a Usted, con la finalidad de hacer de su conocimiento que el Sr. **CHRISTIAN ERICK GONZALES GONZALES**. De la Escuela de Ingeniería de Sistemas de la Institución Universitaria que Usted representa, ha sido admitido para realizar su investigación en el desarrollo de su Tesis denominada **"ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP PARA EL RENDIMIENTO DE APLICACIONES WEB"** en nuestra empresa.

Aprovecho la oportunidad para expresarle mi consideración y estima personal.

Atentamente,



ANTONIO GONZALES BALLENA  
GERENTE DE LA EMPRESA

## ANEXO Nº 2 – PERMISO PARA LA RECOLECCIÓN DE LOS DATOS.

**SOLICITO: PERMISO PARA LA RECOLECCIÓN DE LOS DATOS.**

**SR. GERENTE**

ANTONIO GONZALES BALLENA

Yo **CHRISTIAN ERICK GONZALES GONZALES** identificado con DNI Nº **73334419** con domicilio en Calle Tacna Nº 768, distrito de Monsefú Provincia de Chiclayo. Ante usted respetuosamente me presento y expongo:

Que siendo estudiante del décimo ciclo de la carrera Profesional de Ingeniería de Sistemas de la Universidad Señor de Sipan solicito a Usted me conceda el permiso para la recolección de datos para mi proyecto de investigación titulado "ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP PARA EL RENDIMIENTO DE APLICACIONES WEB" ya que es necesario para la presentación de informe final.

Por lo expuesto:

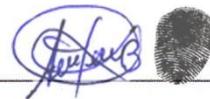
Ruego a usted acceder a mi petición.

Monsefú, Jueves 04 de abril del 2019.



Firma del Investigador

Christian Erick Gonzales Gonzales



Firma del Gerente Pollería Richard

Antonio Gonzales Ballena

**ANEXO Nº 3 – CONSENTIMIENTO INFORMADO DE RECOLECCIÓN DE DATOS.**



POLLERÍA "RICHARD"

**CONSENTIMIENTO INFORMADO DE RECOLECCIÓN DE DATOS PARA PROYECTO DE INVESTIGACIÓN.**

**EL GERENTE GENERAL DE LA POLLERÍA "RICHARD", QUE SUSCRIBE Y OTORGA LA PRESENTE:**

Se solicita dar consentimiento para que el Sr. CHRISTIAN ERICK GONZALES GONZALES identificado con DNI N°73334419 con domicilio en Calle Tacna N° 768 distrito de Monsefú Provincia de Chiclayo participe en el estudio de investigación titulado: ANÁLISIS COMPARATIVO DE PATRONES DE DISEÑO MVC Y MVP PARA EL RENDIMIENTO DE APLICACIONES WEB. El estudio de investigación incluirá: (recolección de datos, etc.), en el cual autorizo voluntariamente para que lleve a cabo su proyecto de investigación en esta empresa.

Por lo expuesto:

Acceder a mi petición.

Monsefú, Lunes 08 de abril del 2019.

Firma del Gerente  
Antonio Gonzales Ballena

08/04/2019

Fecha

Nombre del investigador que obtiene el consentimiento:

CHRISTIAN ERICK GONZALES GONZALES

Firma del Investigador

08/04/2019

Fecha

Av. 7 de junio N° 563 - MONSEFÚ