



**Universidad
Tecnológica
del Perú**

CURSO INTEGRADOR I: SISTEMAS – SOFTWARE

**“SISTEMA WEB DE GESTIÓN DE COMERCIO ELECTRÓNICO PARA BODEGAS EN
LIMA NORTE”**

PROFESOR:

GUEVARA JIMENEZ JORGE ALFREDO

INTEGRANTES:

- | | |
|---------------------------------------|-----------|
| • CARRASCO TAMARIZ JEAN PIER ANDERSON | U21313272 |
| • SANCHEZ LUYO BRYAN | U19213512 |
| • MÁRQUEZ FERNÁNDEZ DIEGO RAÚL | U20306282 |

2023

INDICE DE FIGURAS

Figura 1. Modelo Lean Canva	17
Figura 2. Project Charter	18
Figura 3. Work Breakdown Structure de BodegaFast.....	19
Figura 4. Diagrama de Gantt de BodegaFast	20
Figura 5. Producto Minimo Viable de BodegaFast.....	21
Figura 6. Diagrama de caso de uso de BodegaFast.....	23
Figura 7. Tablas de BodegaFast.....	24
Figura 8. Diagrama de Clase de BodegaFast.....	25
Figura 9. Diseño del registro del usuario (Modelo 1)	26
Figura 10. Diseño del registro del usuario (Modelo 2)	26
Figura 11. Diseño del registro del usuario (Modelo 3).....	27
Figura 12. Diseño del inicio de sesión del usuario (Modelo 1)	27
Figura 13. Diseño del inicio de sesión del usuario (Modelo 2)	28
Figura 14 Diseño del inicio de sesión del usuario (Modelo 3)	28
Figura 15. Dashboard para la administración de las marcas de los productos (Modelo 1).....	29
Figura 16. Dashboard para la administración de las marcas de los productos (Modelo 2).....	29
Figura 17. Dashboard para la administración de las marcas de los productos (Modelo 3).....	30
Figura 18. Dashboard para la administración de las categorías de los productos (Modelo 1)...	31
Figura 19. Dashboard para la administración de las categorías de los productos (Modelo 2)...	31
Figura 20. Dashboard para la administración de las categorías de los productos (Modelo 3)...	32

Figura 21. Dashboard para la administración de bodegas (Modelo 1)	33
.....	33
Figura 23. Dashboard para la administración de bodegas (Modelo 2)	34
Figura 24. Dashboard para la administración de bodegas (Modelo 2 modal)	34
Figura 25. Dashboard para la administración de bodegas (Modelo 3)	35
Figura 26. Dashboard para la administración de productos para las bodegas (Modelo 1)	35
Figura 27. Dashboard para la administración de productos para las bodegas (Modelo 1 modal)	36
Figura 28. Dashboard para la administración de productos para las bodegas (Modelo 2)	36
Figura 29. Dashboard para la administración de productos para las bodegas (Modelo 2 modal)	37
Figura 30. Dashboard para la administración de productos para las bodegas (Modelo 3)	37
Figura 31. Página de inicio del comercio electrónico (Modelo 1)	38
Figura 32. Página de inicio del comercio electrónico (Modelo 2)	38
Figura 33. Página de inicio del comercio electrónico (Modelo 3)	39
Figura 34. Agregar producto (Primer paso)	39
Figura 35. Agregar producto (Segundo paso)	40
Figura 36. Agregar producto (Tercer paso)	40
Figura 37. Flujo de navegacion de pantalla de BodegaFast	41
Figura 38. Diagrama de procesos de BodegaFast	43
Figura 39. Diagrama de entidad relacion fisica de BodegaFast	44
Figura 40. Script del diagrama entidad relacion de BodegaFast	45

Figura 41. Registro de productos de BodegaFast.....	46
Figura 42. Mostrar productos de BodegaFast.....	46
Figura 43. Mostrar productos de clientes de BodegaFast.....	47
Figura 44. Actualizar productos de BodegaFast	47
Figura 45. Eliminar productos de BodegaFast.....	48
Figura 46. Vista principal	48
Figura 47. Vista iniciar sesión y registro	49
Figura 48. Vista agregar bodega	49
Figura 49. Vista agregar categorías.....	50
Figura 50. Vista agregar marcas	50
Figura 51. Vista agregar productos.....	51
Figura 52. Vista mis productos	51
Figura 53. Carrito de Compras (Primer paso).....	52
Figura 54. Carrito de Compras (Segundo paso)	53
Figura 55. Carrito de Compras (Tercer paso)	53
Figura 56. Carrito de Compras (Modelo 2)	54
Figura 57. GitHub carpeta y entornos de desarrollo	54
Figura 58. Estructura de carpetas	55
Figura 59. Ramas de desarrolladores y rama Master	55
Figura 60. Código de Base de datos listar productos de compra.....	67
Figura 61. Código de Modelo para listar productos de compra.....	68

Figura 63. Código de Controlador para listar productos de compra	69
Figura 64. Código de Vista para listar productos de compra.....	70
Figura 67. DA - ERROR	73
Figura 65. DA - NORMAL	73
Figura 66. RE - NORMAL.....	73
Figura 68. RE - ERROR	73
Figura 69. DA - NORMAL	74
Figura 70. RE - NORMAL.....	74
Figura 71. DA - ERROR	74
Figura 72. RE - ERROR	74
Figura 74. RE - NORMAL.....	75
Figura 73. DA - NORMAL	75
Figura 76. RE - ERROR	75
Figura 75. DA - ERROR	75
Figura 77. DA - NORMAL	76
Figura 78. RE - NORMAL.....	76
Figura 80. RE - ERROR	76
Figura 79. DA - ERROR	76
Figura 81. DA - NORMAL	77
Figura 82. RE - NORMAL.....	77
Figura 83. DA - ERROR	77

Figura 84. RE - ERROR	77
Figura 87. DA - ERROR	78
Figura 85. DA - NORMAL	78
Figura 86. RE - NORMAL.....	78
Figura 88. RE - ERROR	78
Figura 89. DA - NORMAL	79
Figura 90. RE - NORMAL.....	79
Figura 91. DA - ERROR	79
Figura 92. RE - ERROR	79
Figura 93. DA - NORMAL	80
Figura 94. RE - NORMAL.....	80
Figura 96. RE - ERROR	80
Figura 95. DA - ERROR	80
Figura 97. Venta de productos para bodega (Primer paso)	81
Figura 98. Venta de productos para bodega (Segundo paso).....	81
Figura 99. Venta de productos para bodega (Tercer paso)	82
Figura 100. Venta de productos para bodega (Modelo 2)	82
Figura 101. GitHub del proyecto..... ¡Error! Marcador no definido.	
Figura 102. Colocacion del archivo phar.php.....	84
Figura 104. Ejecucion del archivo phar.php	85
Figura 105. Creacion del archivo ejecutable bodegafast.php	85

Figura 106. Copia del archivo en una nueva maquina.....	86
Figura 107. Datos del archivo descomprimido.php	86
Figura 108. Datos del archivo phar.php	87
Figura 110. Configuracion del Host.....	88
Figura 111. Creacion de un nuevo controlador	89
Figura 112. Creacion de la clase correo	89
Figura 113. Funcion de agregar ventas	90
Figura 114. Generacion de venta llamando al metodo del correo	90
Figura 115. Comprobante de compra de BodegaFast	92
Figura 116. Metodo de Pago	93
Figura 117. Medio de Pago - fronted	93
Figura 118. Controlador para los métodos de pago	94
Figura 119. Integración de culqi.....	95
Figura 120. Mis ventas para el rol de bodegas.....	95
Figura 121. Reporte detallado de ventas	96
Figura 122. Integración del reporte.....	96
Figura 123. Modulo historial Producto	97
Figura 125. Segundo diseño de las alternativas de Dashboard.....	98
Figura 126. Tercer diseño de las alternativas de Dashboard	99
Figura 127. Sistema de Bodegafast.....	100
Figura 128. Código de creación del grafico de barras	101

Figura 129. Código de creación del grafico del pastel	101
Figura 130. Código de creación del grafico de lineas	102
Figura 134. Código de estructura de capa modelo del dashboard.....	105
Figura 135. Código de estructura del procedimiento almacenado (parte1).....	106
Figura 139. Exportación en PDF del dashboard	110
Figura 140. Exportación de las ventas en formato Excel	111
Figura 141. Descarga desde el navegador el reporte en excel.....	111
Figura 142. Reporte en excel del detalle de ventas	112
Figura 143. Controlador modificado para la exportación en excel	112
Figura 144. Código de estructura del modelo para obtener los datos del reporte	113
Figura 145. Código de estructura del procedimiento de almacenado	113
Figura 146. Código de la vista para exportar datos en Excel.....	114
Figura 147. Mockup de Venta bodega	115
Figura 148. Mockup de Carrito de compra.....	115
Figura 149. Mockup de métodos de pago (parte 1)	116
Figura 150. Mockup de Carrito de compra (parte 2)	116
Figura 151. Tabla de resultados arrojados por PHPUNIT	117

INDICE DE TABLAS

Tabla 1. Tabla de requerimiento funcional de Bodegafast.....	21
Tabla 2. Arquitectura de desarrollo de Bryan	58
Tabla 3. Arquitectura de desarrollo Diego	61
Tabla 4. Arquitectura de desarrollo Jean Pier	64
Tabla 5. Arquitectura de desarrollo en entorno de producción	66
Tabla 6. Tabla de plan de pruebas.....	72
Tabla 7. Tabla de selección de las alternativas de Dashboard	99

INDICE GENERAL

1.	IDEA DE NEGOCIO	14
1.1.	Descripción del problema	14
1.2.	Visión.....	14
1.3.	Misión.....	15
1.4.	Entorno.....	15
1.5.	Estrategias	15
1.6.	Planes de la empresa.....	16
1.7.	Objetivo general	16
1.8.	Objetivos específicos.....	16
1.9.	Lean Canva	17
2.	GESTIÓN DE PROYECTO	18
2.1.	Project Charter	18
2.2.	Work Breakdown Structure (WBS).....	19
2.3.	Diagrama de Gantt	20
3.	INGENIERÍA DE SOFTWARE.....	21
3.1.	Producto Mínimo Viable (PMV).....	21
3.2.	Requerimiento funcional	21
3.3.	Requerimiento no funcional	22
3.4.	Diagrama de caso de uso	23
3.5.	Tablas	24

3.6.	Diagrama de clases	25
3.7.	Mockups	26
3.8.	Mockups (Facilidades de Uso) – 3 pasos – Primer entregable	39
	39	
3.9.	Flujo de navegación de pantallas.....	41
3.10.	Diagrama de procesos.....	43
3.11.	Diagrama entidad relación física.....	44
3.12.	Diagrama de clases de software.....	46
3.13.	Vistas desarrolladas – primer entregable	48
3.14.	Mockups (Facilidades de Uso) – 3 pasos – Segundo entregable.....	52
3.15.	GitHub carpetas y entornos de desarrollo.....	54
3.16.	Arquitectura en capas de Bryan.....	56
3.17.	Arquitectura en capas de Diego.....	59
3.18.	Arquitecturas en capas de Jean Pier	61
3.19.	Arquitectura en producción	65
3.20.	Base de datos listar productos para la compra.....	67
3.21.	Modelo para listar productos de compra	68
3.22.	Controlador para listar productos de compra	69
3.23.	Vista para listar productos de compra	70
3.24.	Revisiones de código.....	71
3.25.	Plan de pruebas	72

4.1.	Mockups (Facilidades de Uso) – 3 pasos – Tercer entregable	81
4.2.	GitHub actualizado	83
4.3.	Creación de un ejecutable con extensión phar para comprimir nuestro proyecto	
	84	
4.4.	Notificación por correo electrónico cada vez que se realiza una compra por parte del cliente o de la bodega.	88
5.1.	Mockup método de pago	93
5.2.	Uso de Culqi como medio de pago – frontend	93
5.3.	Creación de las órdenes en el controlador para los métodos de pago	94
5.4.	Integración de culqi en nuestro método “agregarCompra” en el controlador	
Cliente	95	
5.5.	Mockup – Mis ventas para el rol bodegas con reportes	95
5.6.	Mockup – Reporte detalle de ventas.....	96
5.7.	Integración del reporte en el controlador Venta.....	96
5.8.	Mockup actualización de stock y precio de venta	97
6.1.	Primer diseño	98
6.2.	Segundo diseño.....	98
6.3.	Tercer diseño	99
6.4.	Selección de tablero	99
6.5.	Diseño del dashboard para bodegas – Sistema Bodegafast.....	100
6.6.	Exportar en pdf – dashboard	110
6.7.	Exportación de ventas en formato Excel.....	111

6.8.	Mockups adicionales	115
6.9.	Evaluación de la cobertura de código mediante PHPUNIT	117

1. IDEA DE NEGOCIO

1.1. Descripción del problema

Actualmente la falta de plataformas informáticas de pedidos de bodegas cercanas en un lugar hace que los clientes no tengan una forma fácil y conveniente de comprar productos de conveniencia sin tener que salir de sus hogares. Esto puede resultar en una pérdida de tiempo y una experiencia de compra incómoda para los consumidores, lo que puede llevarlos a buscar otras opciones de compra. También la poca variedad de métodos de pagos que tienen las bodegas hace que el cliente pierda el interés de compra y tenga que elegir otros métodos más complejos para adquirir un solo producto, además el desconocimiento de bodegas cercanas hace que adquiera sus productos de otros lugares lejanos como los supermercados, perdiendo tiempo y dinero, ya que normalmente los precios de los supermercados son incrementados de su precio común. Es importante que las bodegas cercanas consideren la implementación de un servicio de pedidos mediante una plataforma web para satisfacer las necesidades y expectativas de los clientes y aumentar sus ventas. Además, muchos de los dueños de las bodegas no llevan un control adecuado de sus ventas y productos, lo que hace que pierdan dinero en sus negocios.

1.2. Visión

Consolidarse como una plataforma web popular en donde todas las personas puedan encontrar la facilidad de adquirir productos en un solo lugar, sin tener en sus dispositivos una aplicación para el supermercado, otra para librerías, entre otras cosas que se pueden encontrar a una bodega.

1.3. Misión

Nuestra misión es impulsar el desarrollo económico en las bodegas locales acelerando la implementación de comercios electrónicos. Además, se busca dar acceso a una gran variedad de productos para los consumidores de manera rápida, fácil y segura.

1.4. Entorno

El entorno abarca los 8 distritos de Lima norte, considerando a toda la población y a las bodegas de estos lugares.

1.5. Estrategias

La estrategia de fidelización que tendrá BodegaFast serán las siguientes:

- Implementar una campaña de promoción de marketing digital para dar a conocer el servicio de comercio electrónico para las bodegas de Lima Norte y atraer a más clientes.
- Establecer precios competitivos y atractivos para los productos ofrecidos en línea, al mismo tiempo que se asegura un margen de ganancia adecuado.
- Proporcionar una experiencia de compra en línea fácil y cómoda, y un servicio de entrega confiable y puntual para satisfacer las necesidades de los clientes.
- Implementar herramientas y tecnologías para optimizar la gestión de pedidos y entregas, incluyendo sistemas de automatización de procesos.
- Considerar la posibilidad de expandir el servicio de comercio electrónico para las bodegas a nuevos productos y áreas geográficas para atraer a más clientes y aumentar las ventas.

1.6. Planes de la empresa

- Crear una impresión amigable y de confianza con los clientes mediante la responsabilidad al momento de la entrega y puntualidad del servicio para aumentar la preferencia de la página web.
- Brindar facilidades a las bodegas de Lima Norte para que sus productos puedan llegar a más lugares y promover el reconocimiento de estas mismas para captar clientes.
- Ser cuidadosos al momento de contratar repartidores, promoviendo el uso de automóviles para disminuir riesgo de accidentes y exigir seguros personales para el caso de que los haya.

1.7. Objetivo general

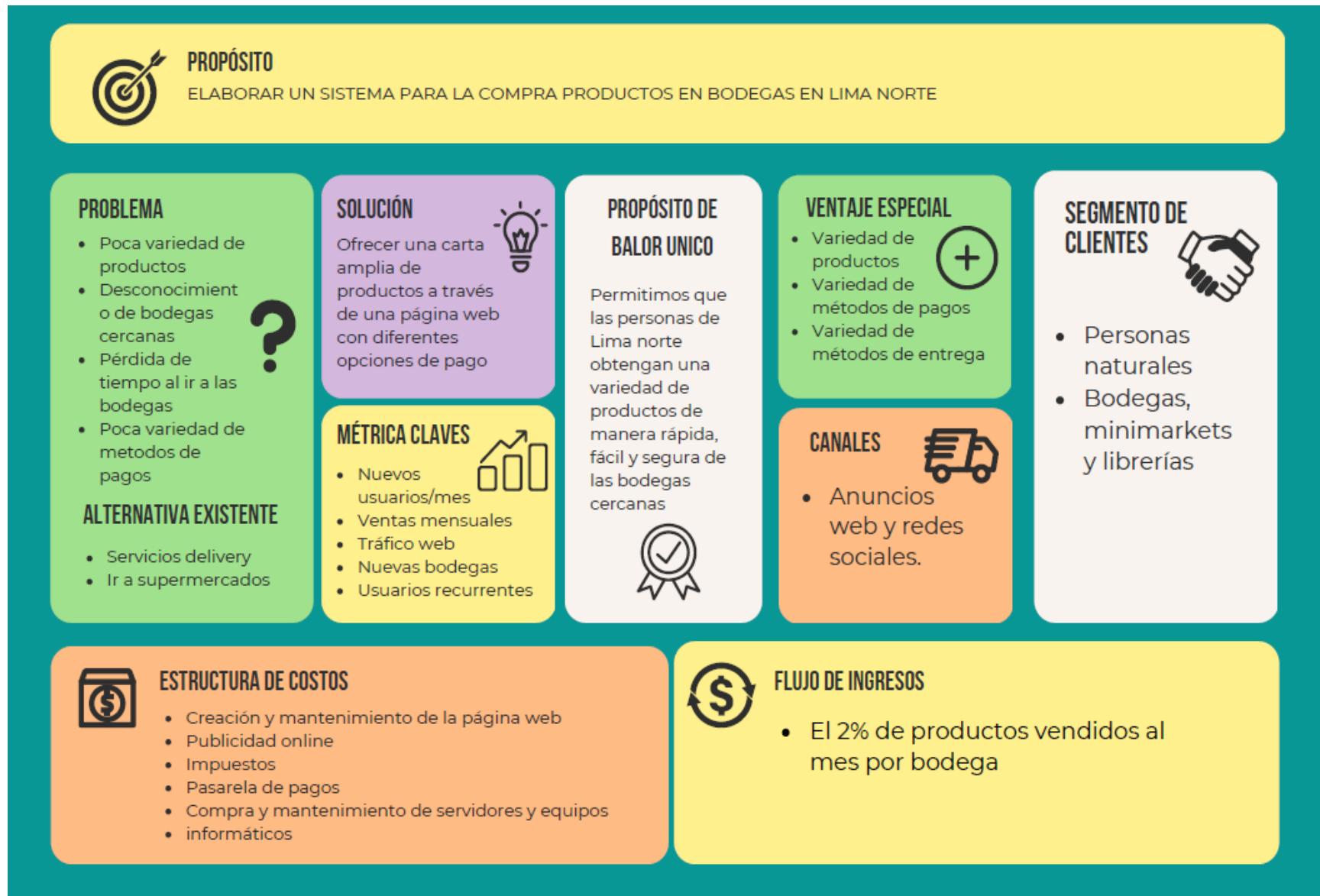
El objetivo principal es desarrollar un sistema web para la gestión del comercio electrónico para bodegas de Lima Norte, captando la atención y lealtad de los ciudadanos de las principales áreas urbanas haciendo frente a la creciente competencia y liderando el Mercado de las páginas web que se dedican a este rubro en del sector.

1.8. Objetivos específicos

- Aumentar el número de clientes que realizan pedidos en línea y entregas a domicilio.
- Mejorar la experiencia del cliente al permitir una forma más fácil y conveniente de comprar productos de conveniencia.
- Optimizar la gestión de pedidos y entregas a través de herramientas de automatización y logística.
- Ampliar el alcance y la visibilidad de las bodegas a través de la presencia en línea y la promoción de productos.
- Recopilar datos de los clientes y estadísticas de pedidos para mejorar la eficiencia y personalización del servicio de pedidos de productos de bodegas.

1.9. Lean Canva

Figura 1. Modelo Lean Canva



2. GESTIÓN DE PROYECTO

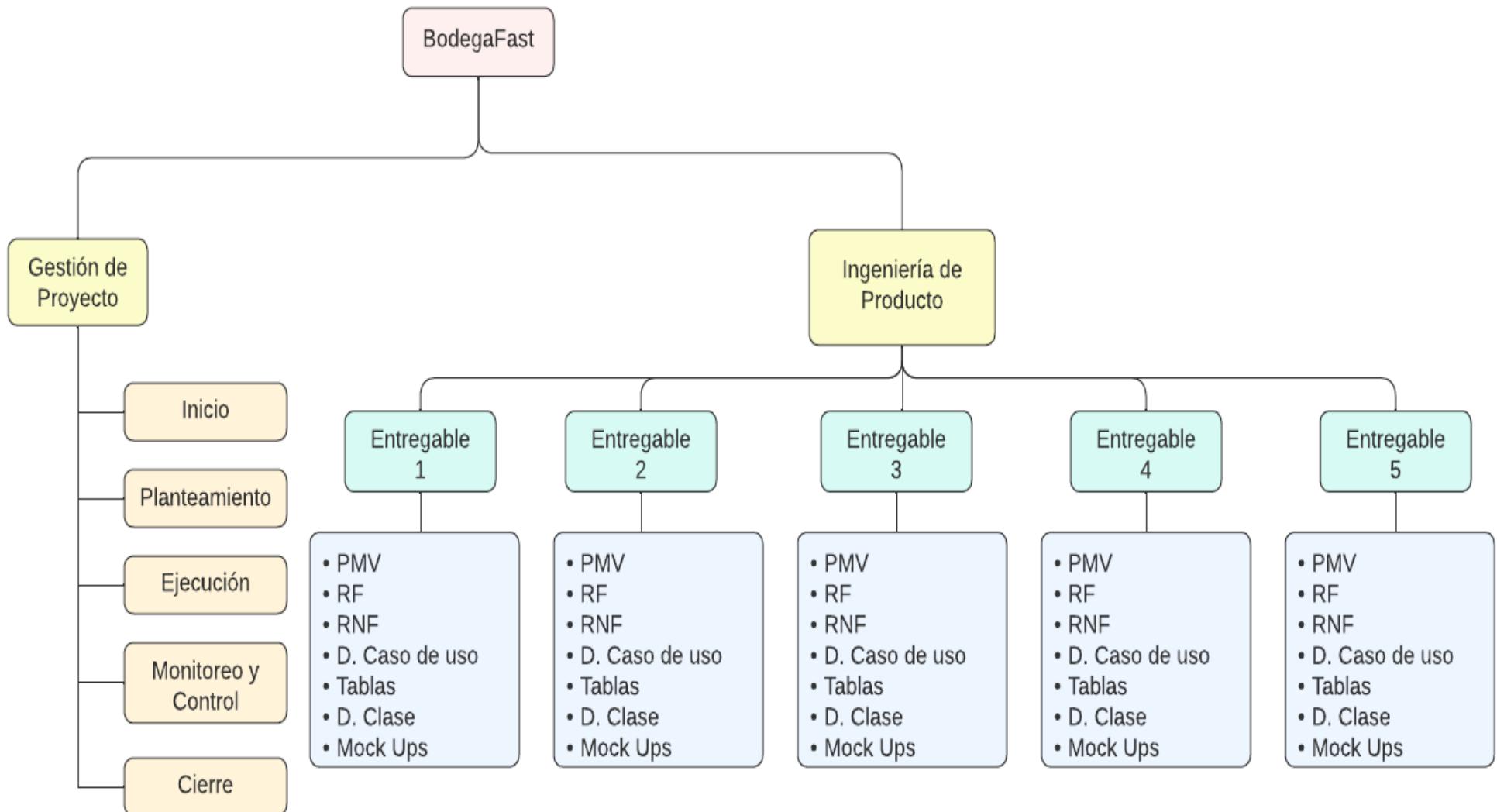
2.1. Project Charter

Figura 2. Project Charter

Project Charter					
Titulo	Sistema de Gestión de Comercio Electrónico para Bodegas	Jefe de Proyecto	Jean Pier		
Fecha inicio	27/03/2023	Fecha fin	24/07/2023		
Necesidad del negocio					
Desarrollar un sistema web de manera que el cliente pueda elegir sus productos y comprarlos con varios métodos de pago y entregas a domicilio, además de contar con módulos de administración, de manera que cada bodega podrá agregar sus productos y administrar sus ventas. El objetivo de este sistema es ayudar a las bodegas a que incrementen sus ganancias y a los clientes a que puedan comprar sus productos de manera rápida, fácil y segura.					
Alcance		Entregables			
Aplicación Web desarrollada en Java		1. Catalogo de productos			
Incluyendo todas las características previstas		2. Ventas de productos (cliente)			
		3. Ventas de productos (bodega)			
		4. Métodos de pago y envío			
		5. Administración de ventas (bodega)			
Riesgos y problemas		Suposiciones y dependencias			
Integrantes del equipo abandonen el proyecto, no cumplir con los entregables a tiempo, complejidad del lenguaje de programación orientado a Web.		Todos los miembros del equipo trabajarán en este proyecto			
Costos					
El sistema desarrollado e implementado tendrá un costo de S/ 380.00					
Cronograma					
Hitos y Actividades		Responsable	Fecha		
1. Catalogo de productos		Diego	27/03/2023		
2. Ventas de productos (cliente)		Jean Pier	24/04/2023		
3. Ventas de productos (bodega)		Bryan	08/05/2023		
4. Métodos de pago y envío		Diego	05/06/2023		
5. Administración de ventas (bodega)		Bryan	26/06/2023		
Equipo		Comité de Aprobación			
Jefe de Proyecto	Jean Pier Carrasco T.	Sponsor			
Supervisor	Diego Márquez F.	Gerente			
Desarrollador	Bryan Sanchez L.	Docente	Jorge Guevara J.		

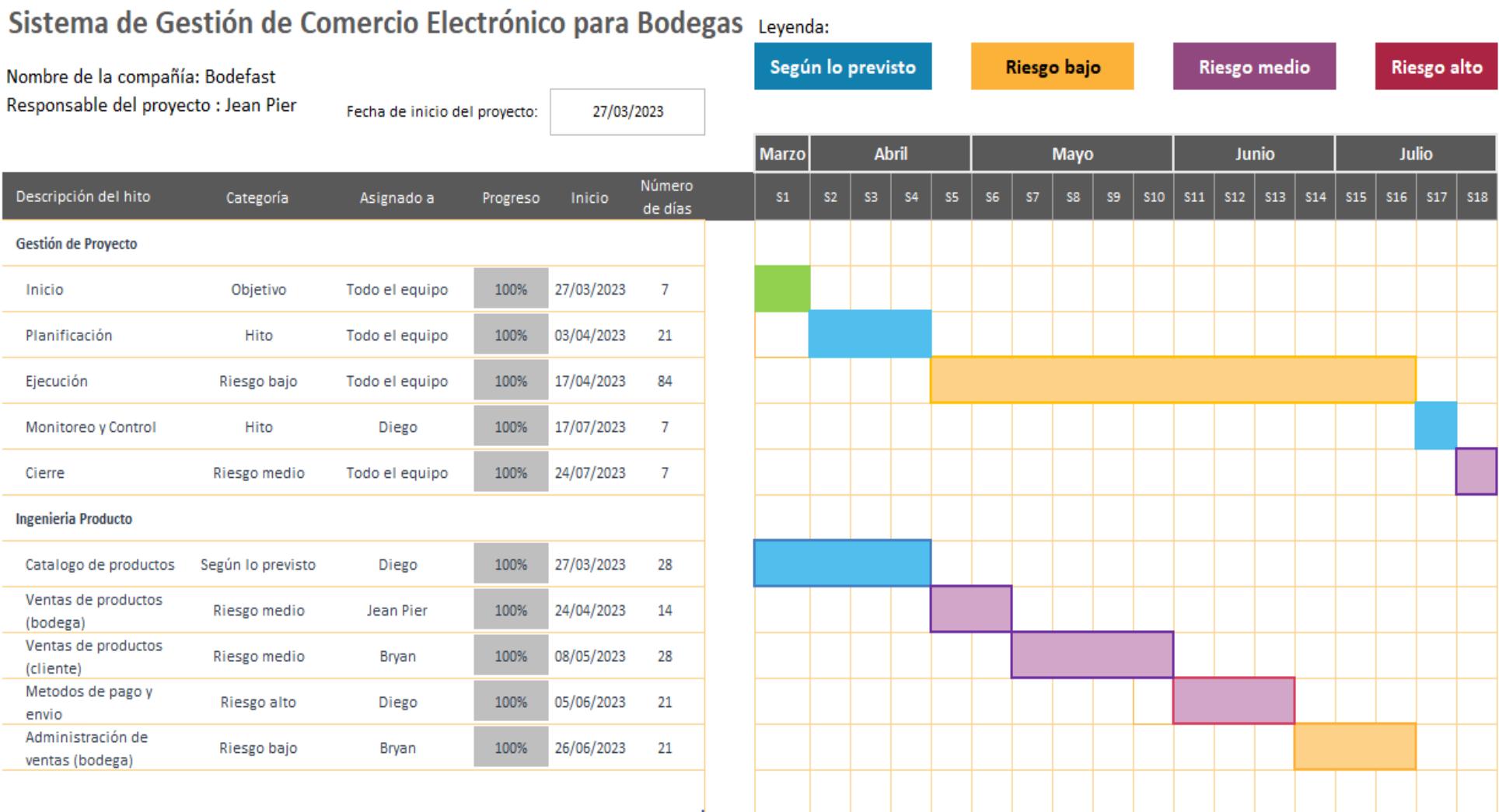
2.2. Work Breakdown Structure (WBS)

Figura 3. Work Breakdown Structure de BodegaFast



2.3. Diagrama de Gantt

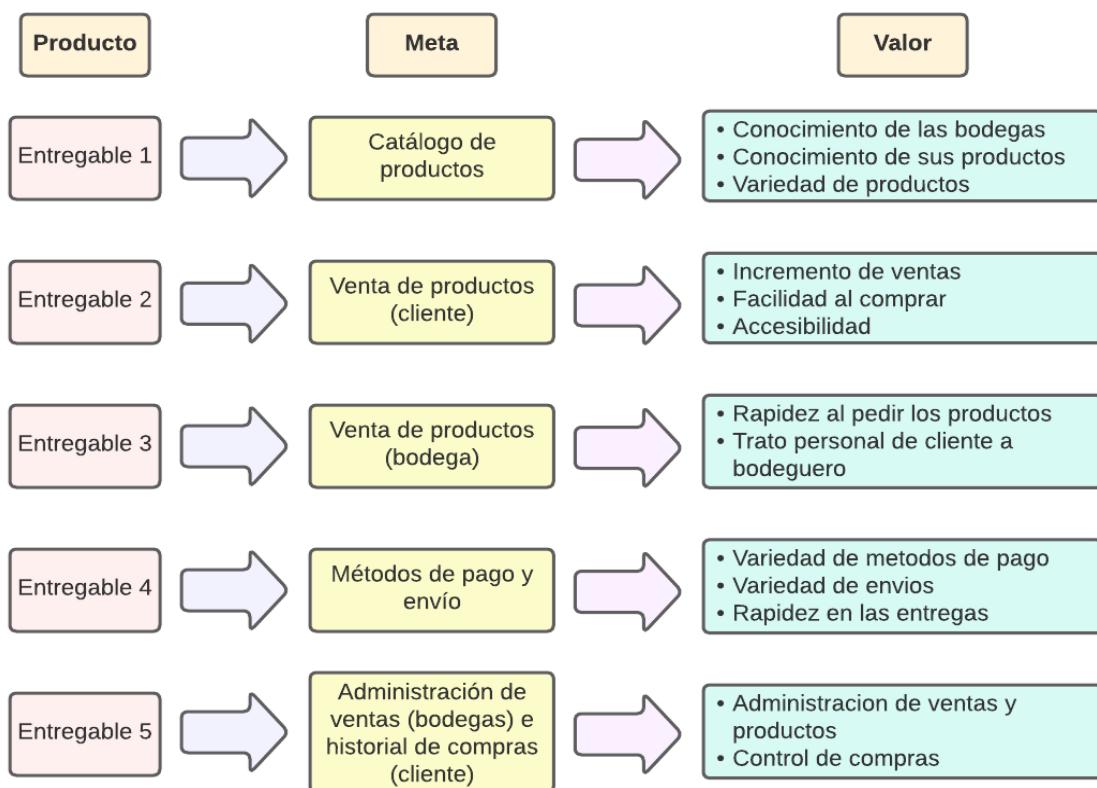
Figura 4. Diagrama de Gantt de BodegaFast



3. INGENIERÍA DE SOFTWARE

3.1. Producto Mínimo Viable (PMV)

Figura 5. Producto Mínimo Viable de BodegaFast



3.2. Requerimiento funcional

Tabla 1. Tabla de requerimiento funcional de Bodegafast

Entidad	Proceso
Usuarios	El sistema permite registrar usuario
	El sistema permite eliminar usuario
	El sistema permite actualizar usuario
	El sistema permite visualizar los productos
Administrador	El sistema permite registrar bodegas

	El sistema permite eliminar bodegas
	El sistema permite actualizar bodegas
	El sistema permite visualizar los productos
	El sistema permite agregar categorías
	El sistema permite actualizar categorías
	El sistema permite eliminar categorías
	El sistema permite agregar marcas
	El sistema permite actualizar marcas
	El sistema permite eliminar marcas
Bodega	El sistema permite registrar productos
	El sistema permite eliminar productos
	El sistema permite actualizar productos
	El sistema permite actualizar bodega
	El sistema permite visualizar los productos

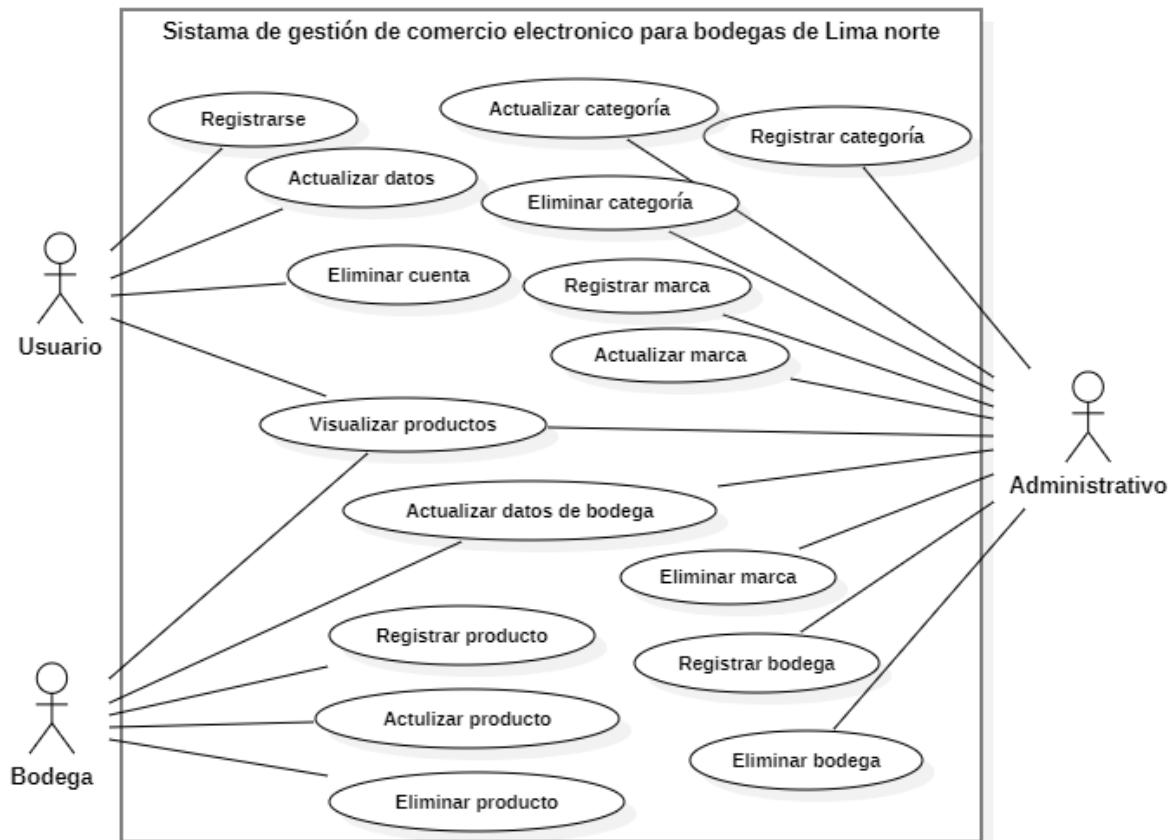
3.3. Requerimiento no funcional

- Para el desarrollo del sistema se implementará con el lenguaje de programación por el lado del cliente HTML 5, CSS3 y JavaScript 6, por el lado del servidor utilizaremos PHP.
- Para la base de datos se utilizará MySql 8.
- Para el editor de código usaremos Visual Studio Code.
- Para el diseño CSS de nuestra página web utilizaremos la librería BootStrap 5.
- Utilizaremos el MySql Workbench como herramienta visual de diseño de base de datos.
- Para los Mock Up se utilizará el software de Figma en su versión más reciente.

- Para elaboración de diagramas se utilizará el software de Star UML.
- El tipo de letra que tendrá el software será “Monserat” que se extraerá de Google Fonts.

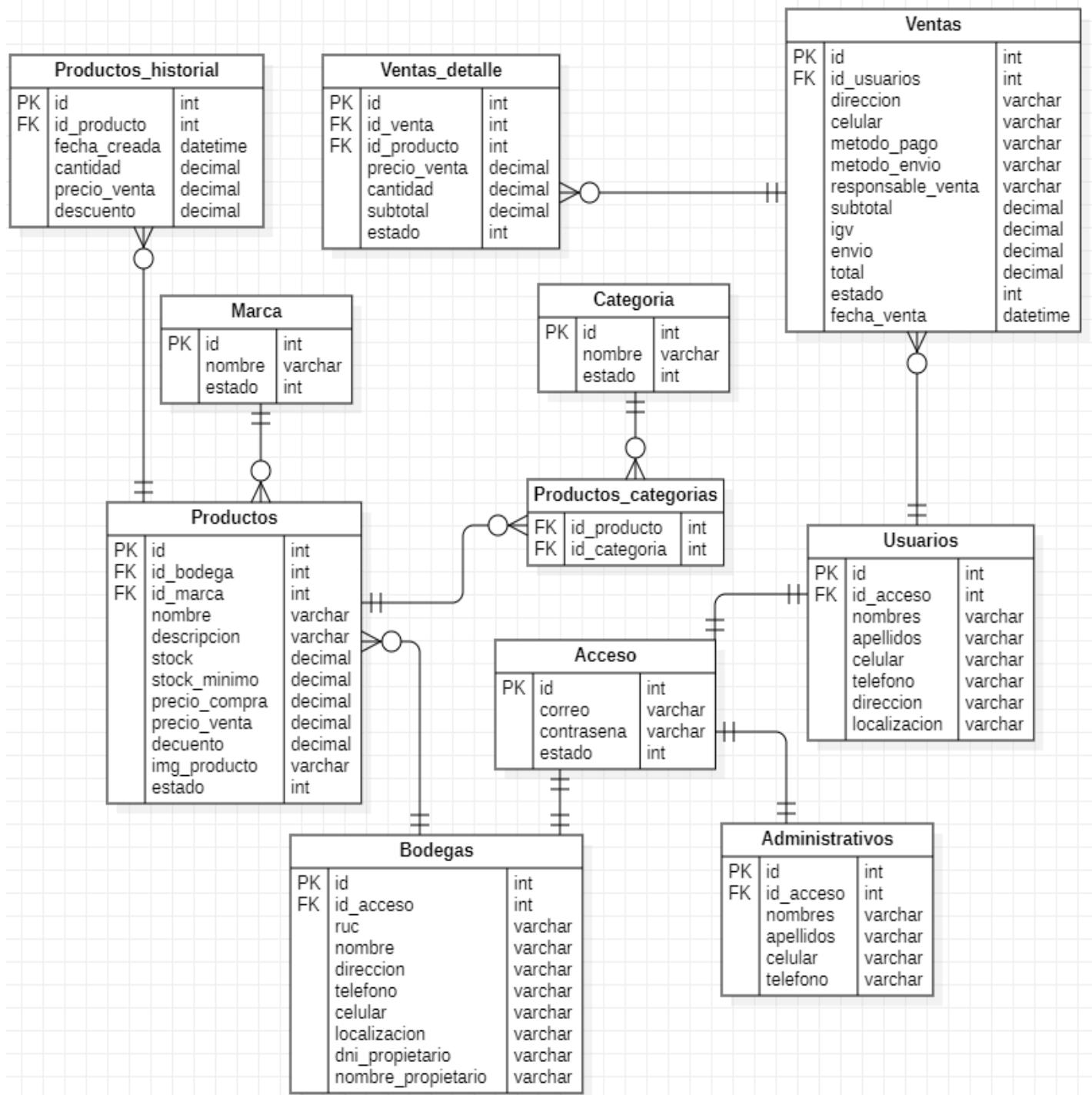
3.4. Diagrama de caso de uso

Figura 6. Diagrama de caso de uso de BodegaFast



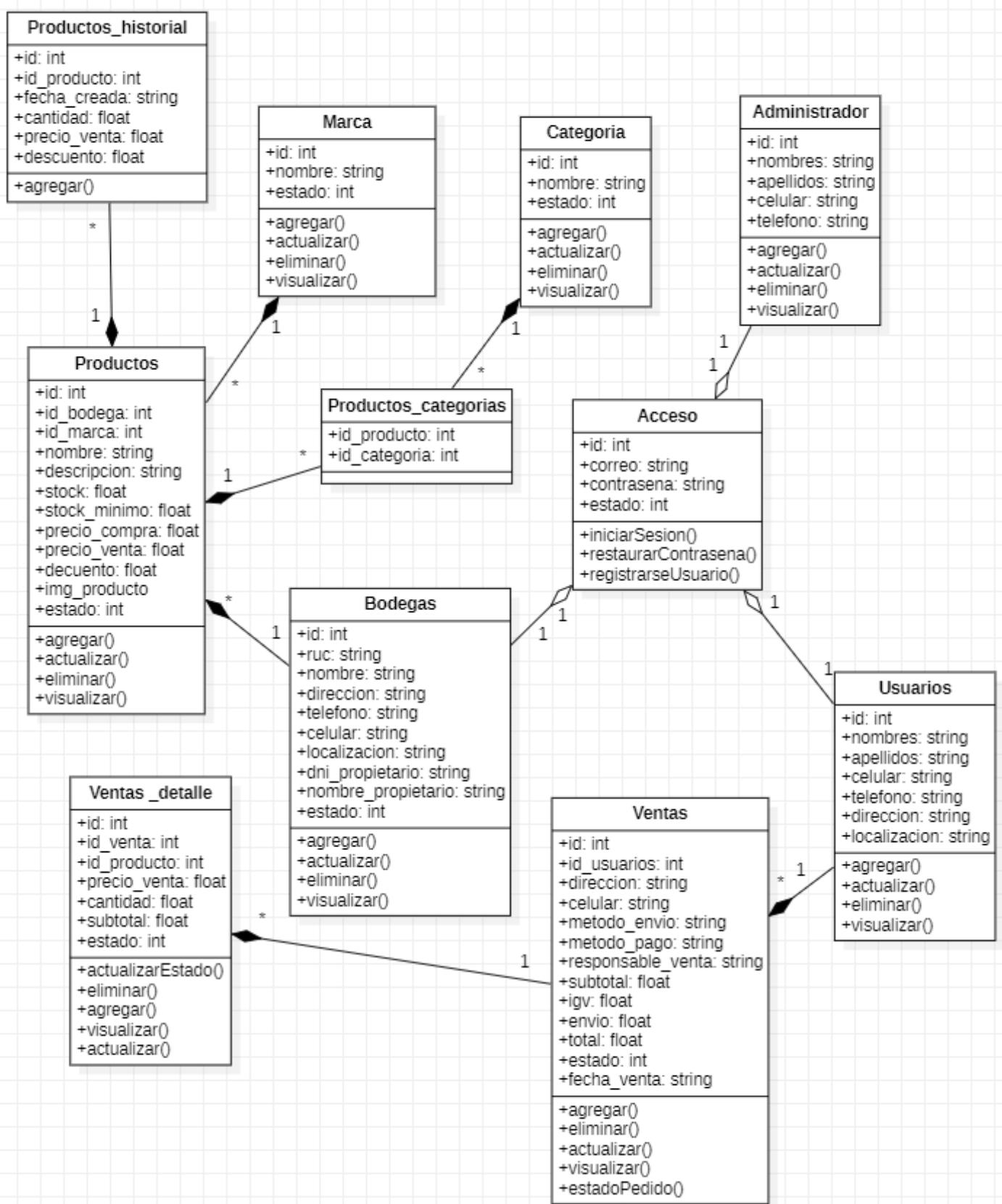
3.5. Tablas

Figura 7. Tablas de BodegaFast



3.6. Diagrama de clases

Figura 8. Diagrama de Clase de BodegaFast



3.7. Mockups

Figura 9. Diseño del registro del usuario (Modelo 1)

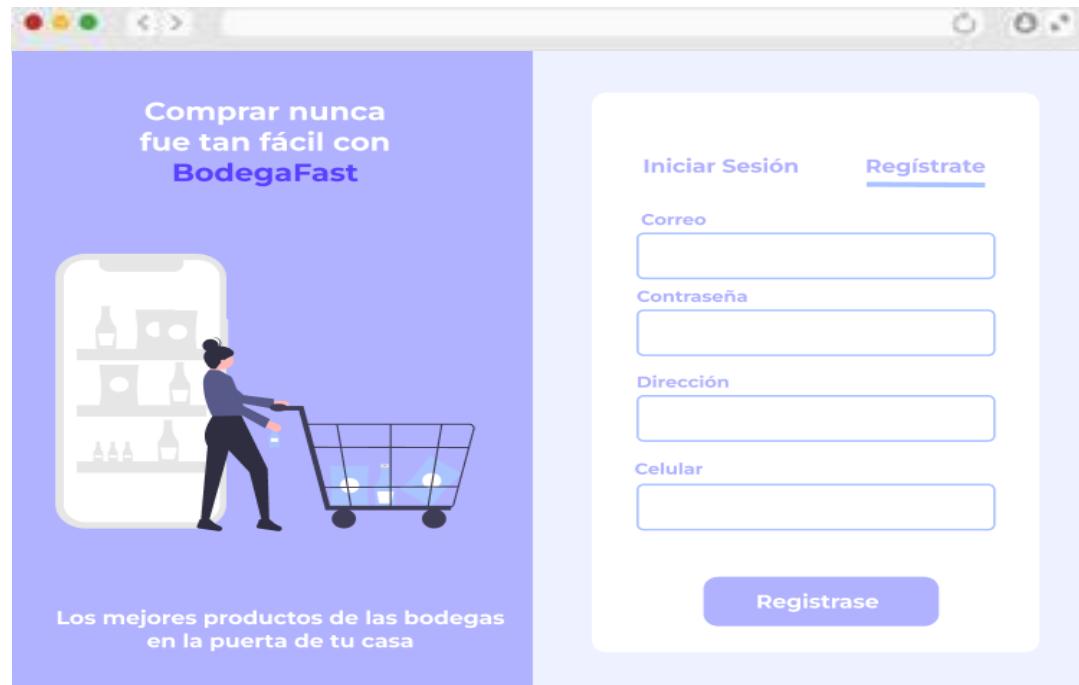


Figura 10. Diseño del registro del usuario (Modelo 2)

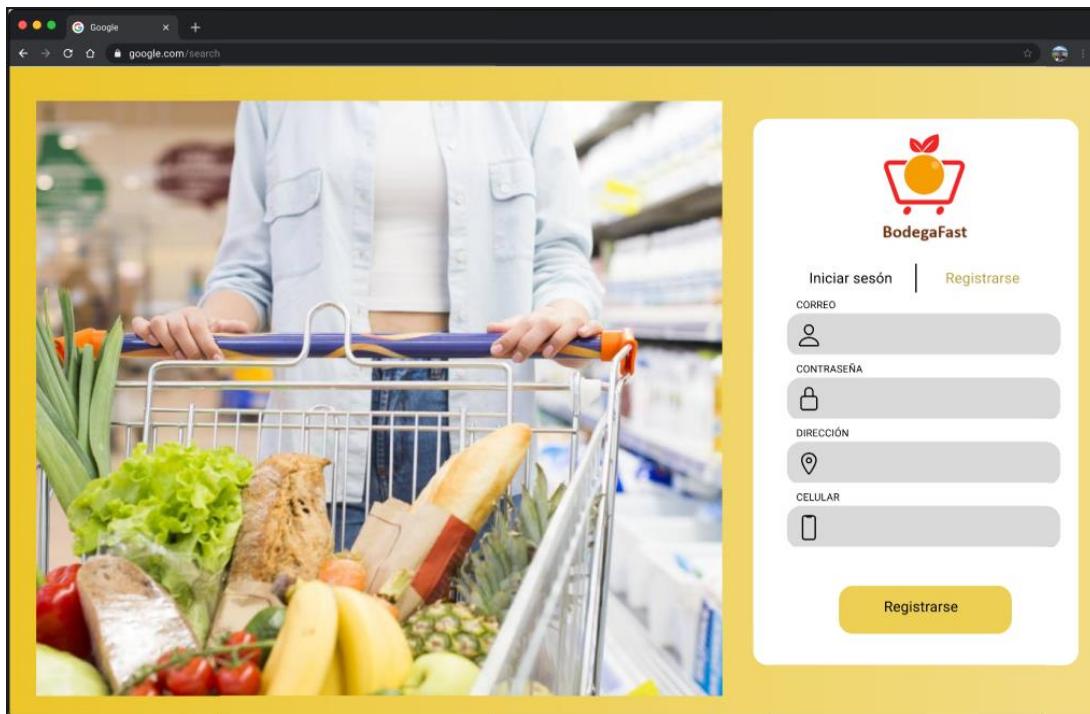


Figura 11. Diseño del registro del usuario (Modelo 3)

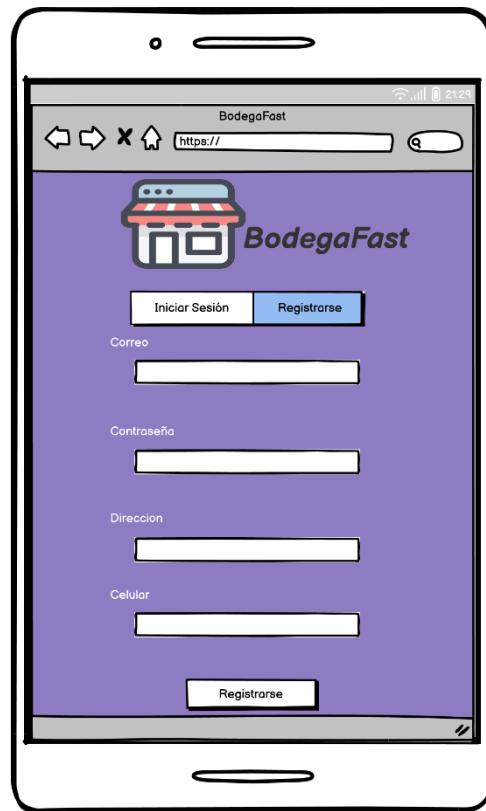


Figura 12. Diseño del inicio de sesión del usuario (Modelo 1)

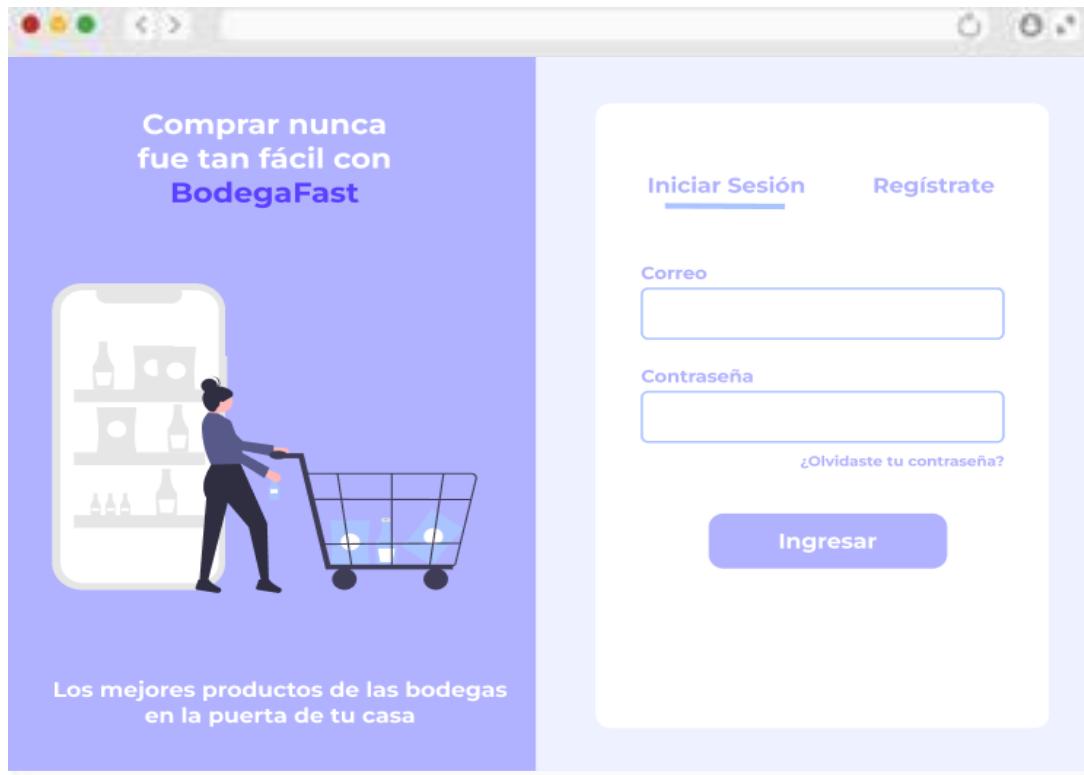


Figura 13. Diseño del inicio de sesión del usuario (Modelo 2)

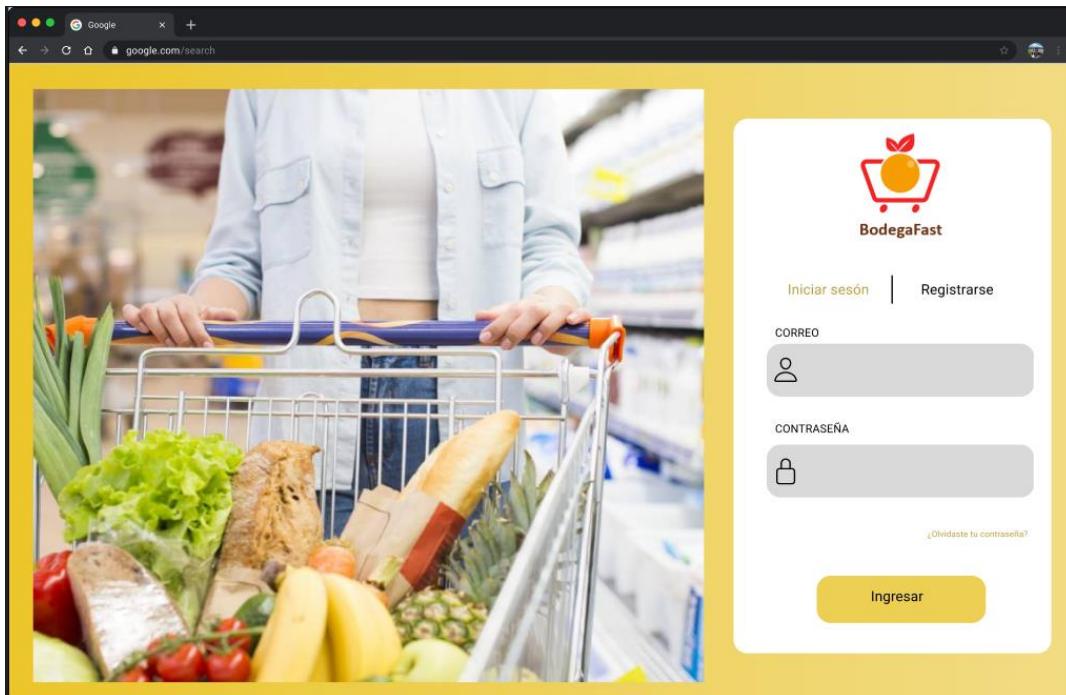


Figura 14 Diseño del inicio de sesión del usuario (Modelo 3)



Figura 15. Dashboard para la administración de las marcas de los productos (Modelo 1)

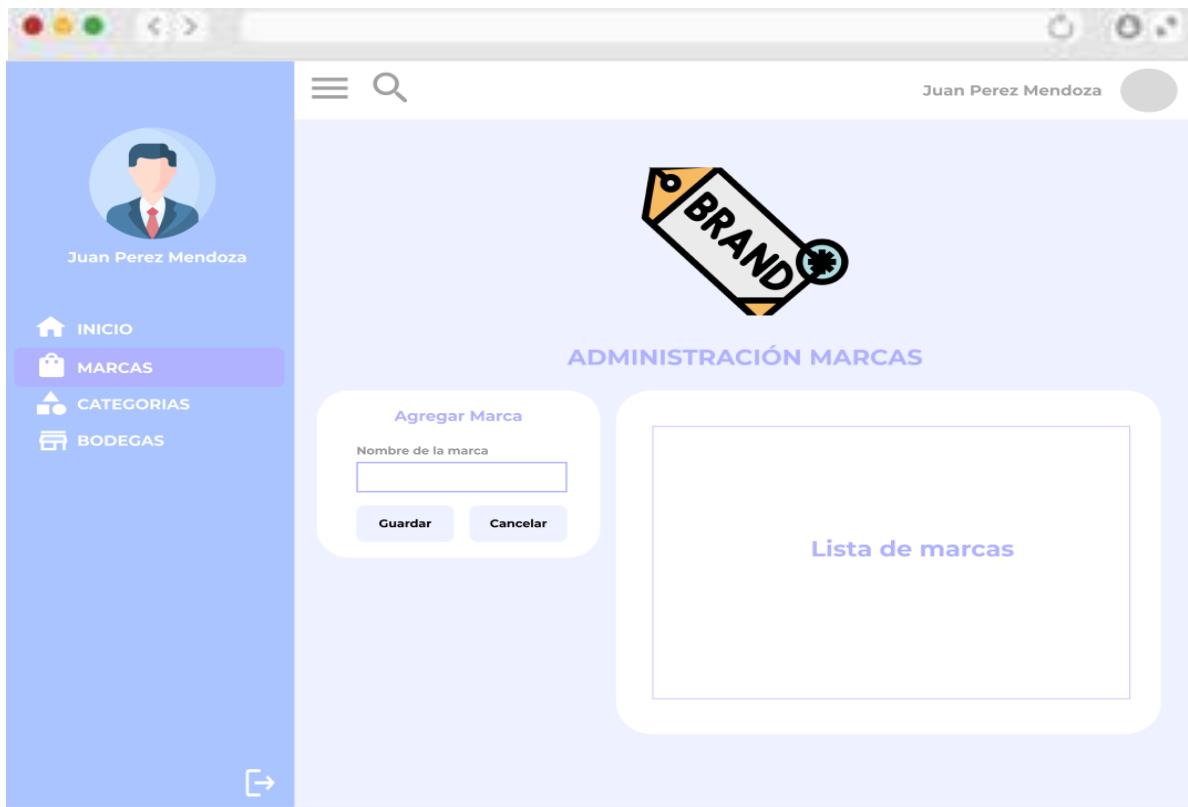


Figura 16. Dashboard para la administración de las marcas de los productos (Modelo 2)

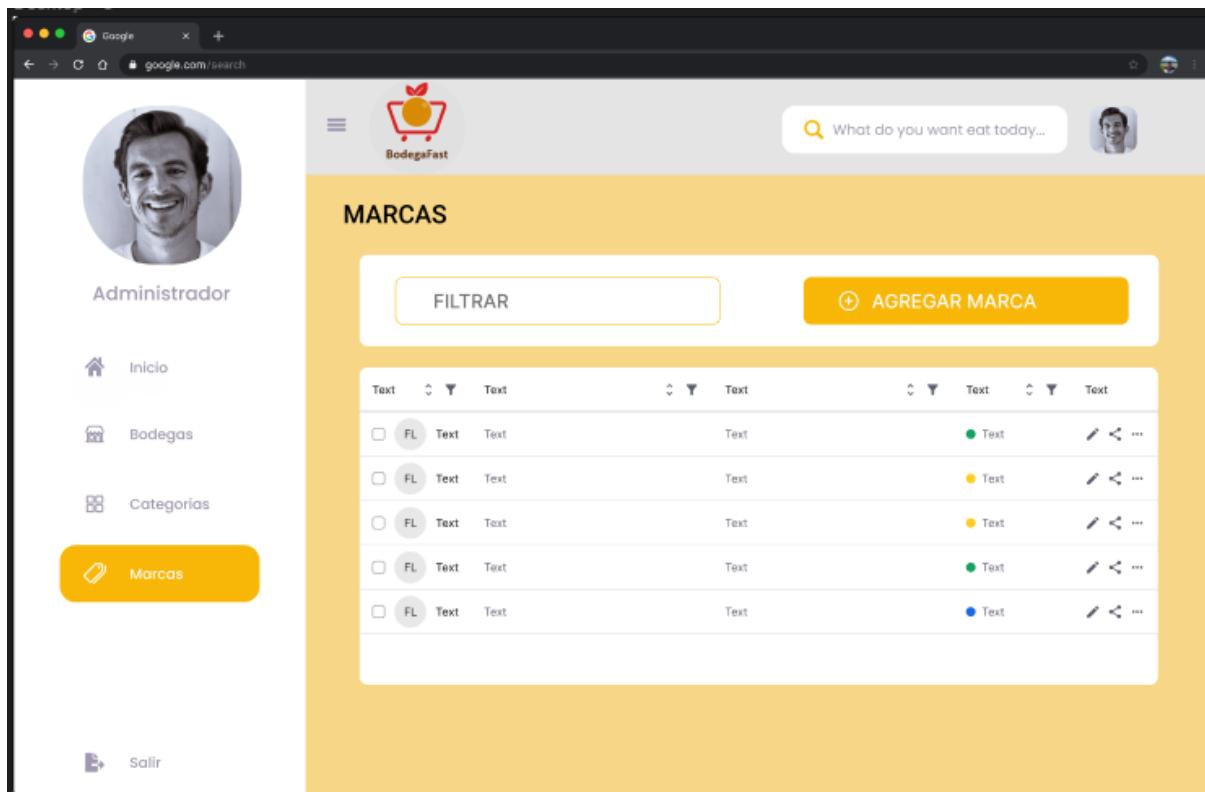


Figura 17. Dashboard para la administración de las marcas de los productos (Modelo 3)



Figura 18. Dashboard para la administración de las categorías de los productos (Modelo 1)

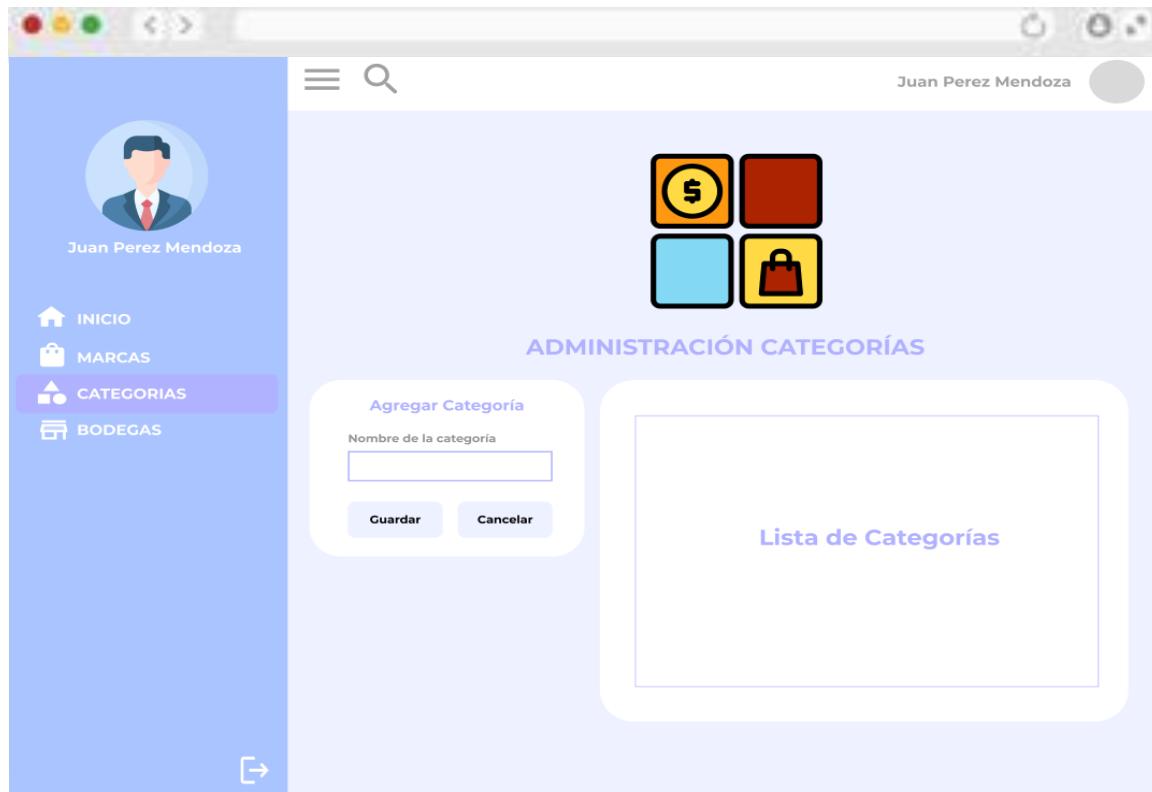


Figura 19. Dashboard para la administración de las categorías de los productos (Modelo 2)

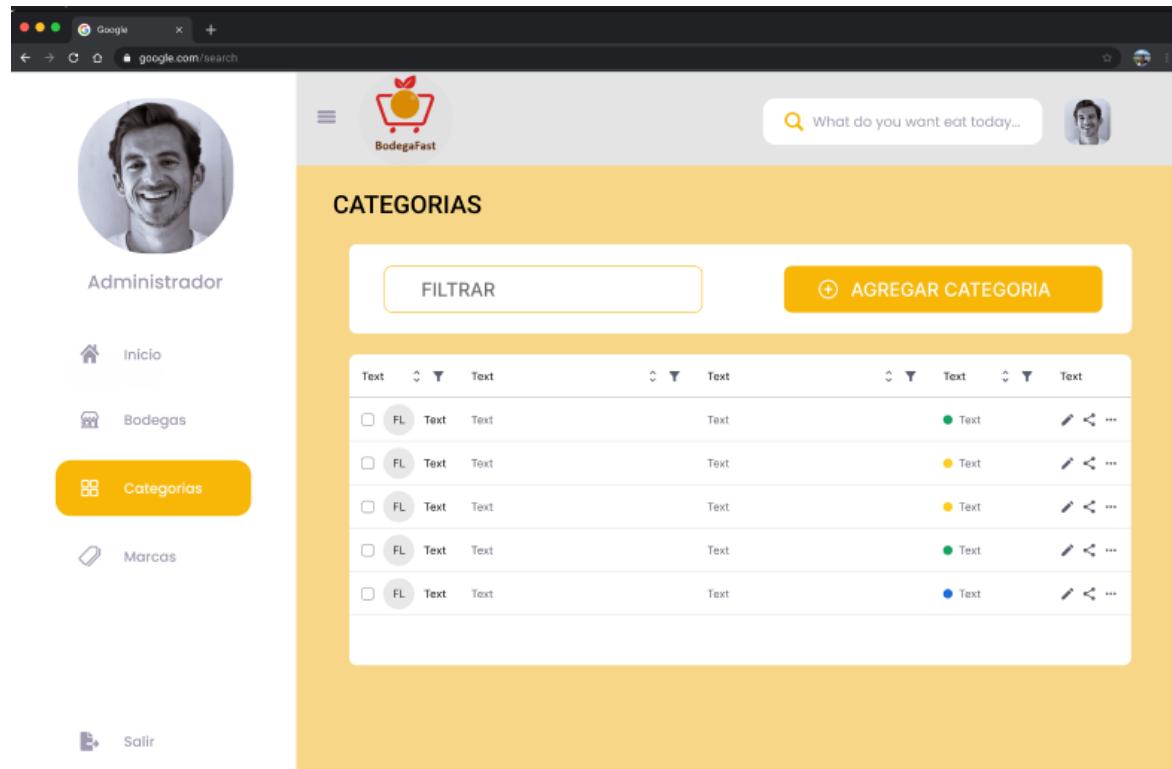


Figura 20. Dashboard para la administración de las categorías de los productos (Modelo 3)



Figura 21. Dashboard para la administración de bodegas (Modelo 1)

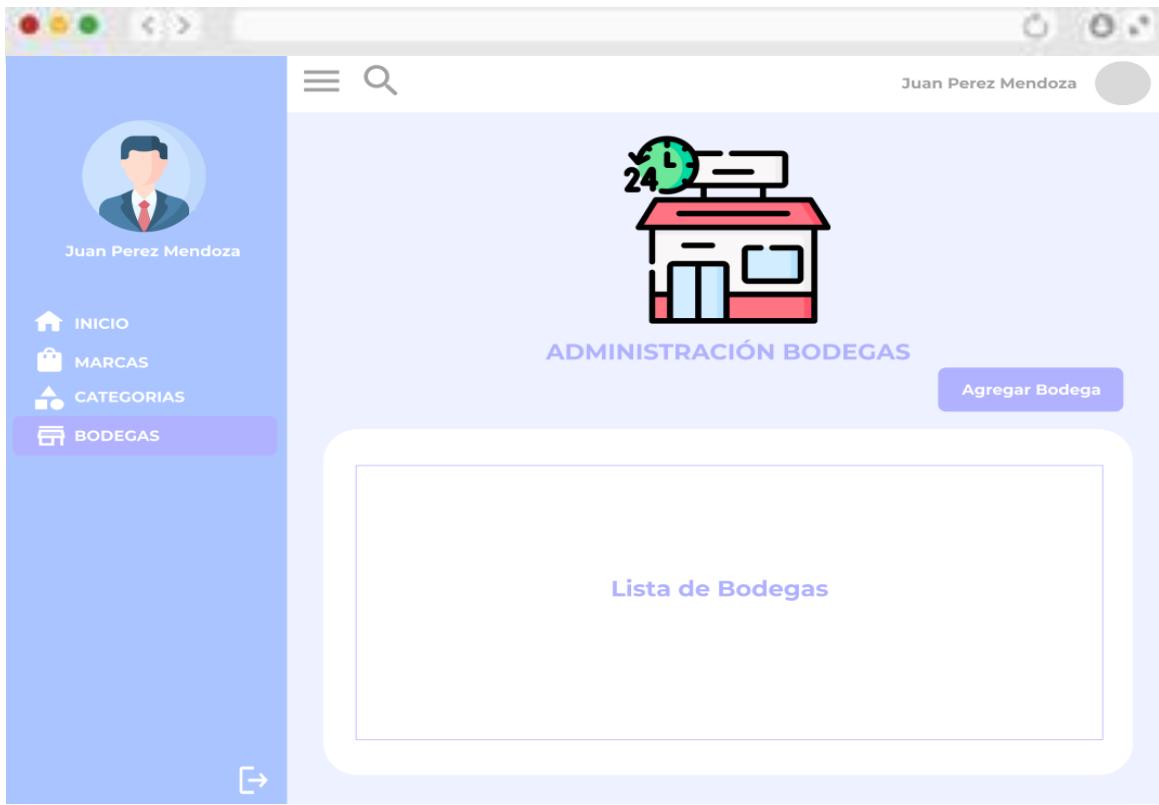


Figura 22. Dashboard para la administración de bodegas (Modelo 1 modal)

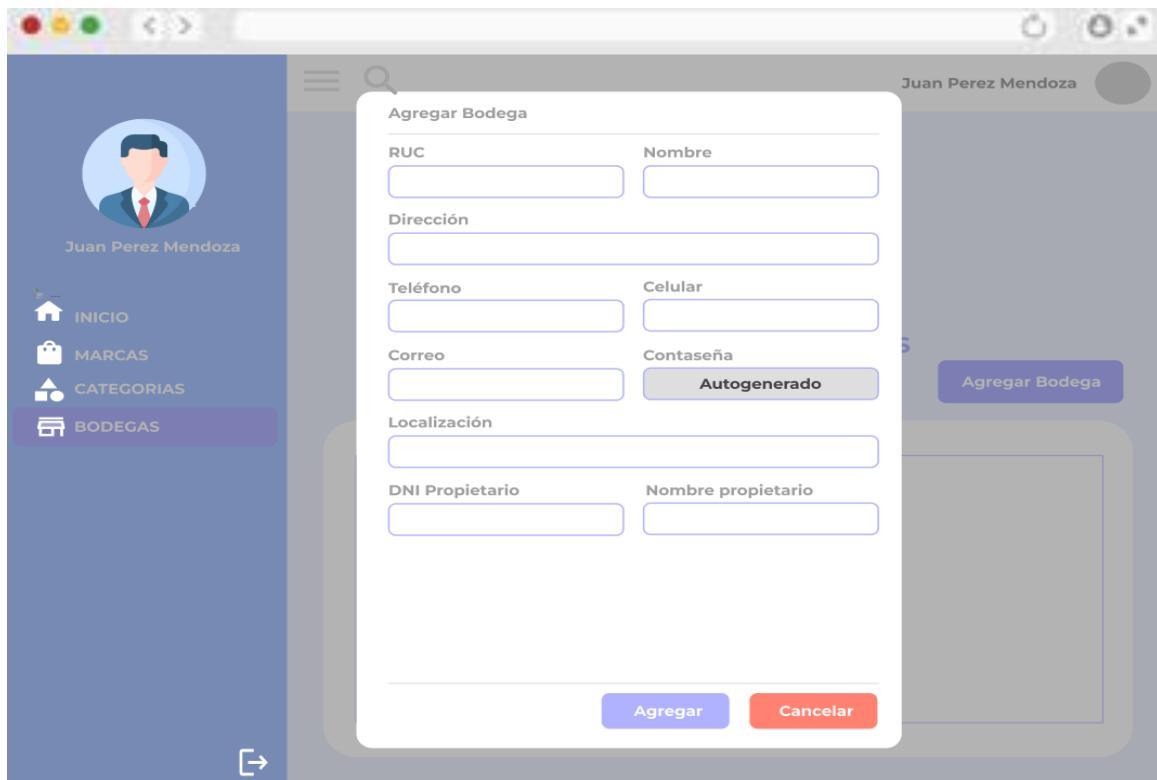


Figura 23. Dashboard para la administración de bodegas (Modelo 2)

The screenshot shows a web-based dashboard for managing warehouses. On the left, a sidebar menu includes 'Inicio', 'Bodegas' (which is highlighted in yellow), 'Categorías', 'Marcas', and 'Salir'. The main area has a yellow header with the title 'BODEGAS'. Below it is a search bar with the placeholder 'What do you want eat today...'. A large button labeled '+ NUEVA BODEGA' is visible. The central part of the screen displays a table with several rows of data, each consisting of a checkbox, a 'FL' icon, a 'Text' column, and another 'Text' column. To the right of the table are icons for edit, delete, and more options.

Figura 24. Dashboard para la administración de bodegas (Modelo 2 modal)

This screenshot shows the same dashboard as Figure 23, but with a modal dialog box overlaid. The dialog is titled 'AGREGAR BODEGA' and contains fields for 'RUC' (with a dropdown menu), 'NOMBRE' (with a dropdown menu), 'DIRECCIÓN' (text input), 'CORREO' (text input), 'TELÉFONO' (text input), 'CELULAR' (text input), and 'CONTRASEÑA' (text input with a dropdown menu showing 'Auto'). At the bottom of the dialog are two buttons: 'Cancelar' and 'Agregar'.

Figura 25. Dashboard para la administración de bodegas (Modelo 3)

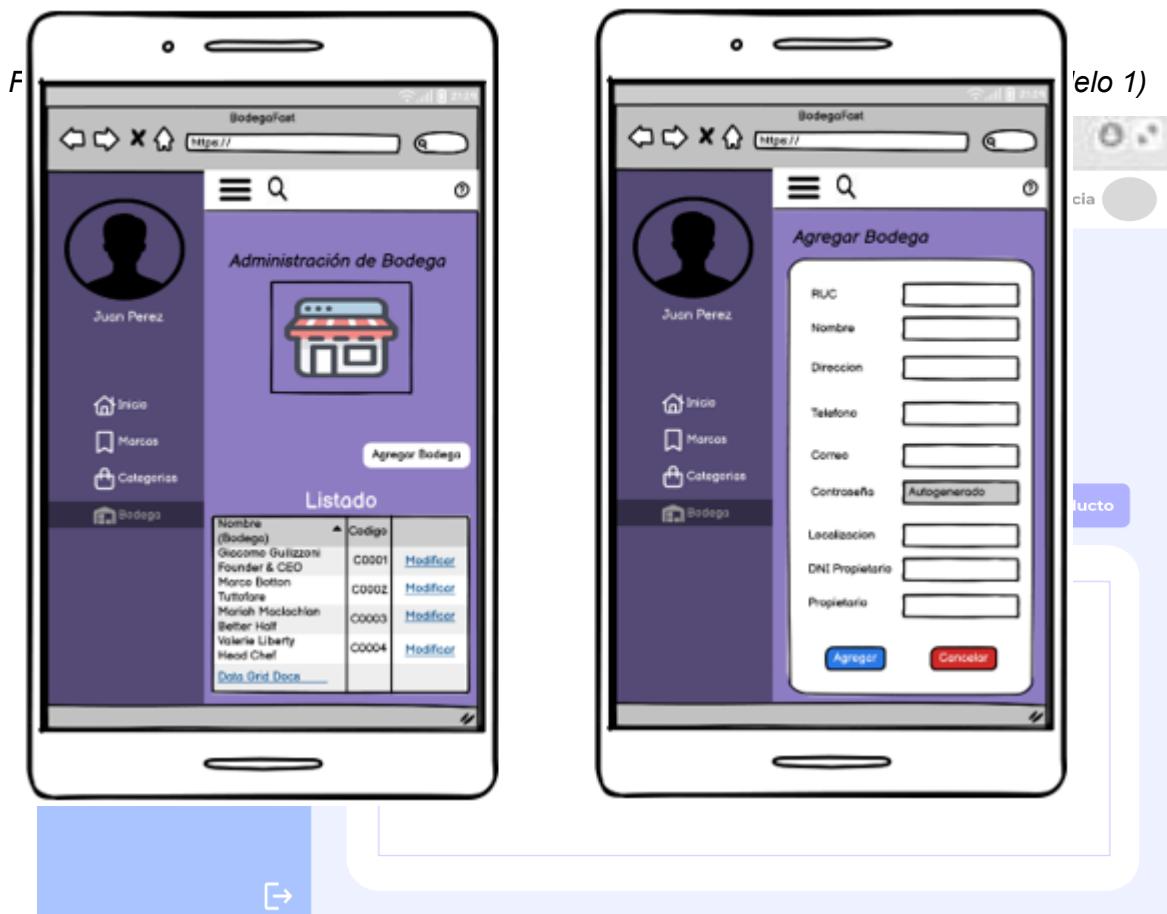


Figura 27. Dashboard para la administración de productos para las bodegas (Modelo 1 modal)

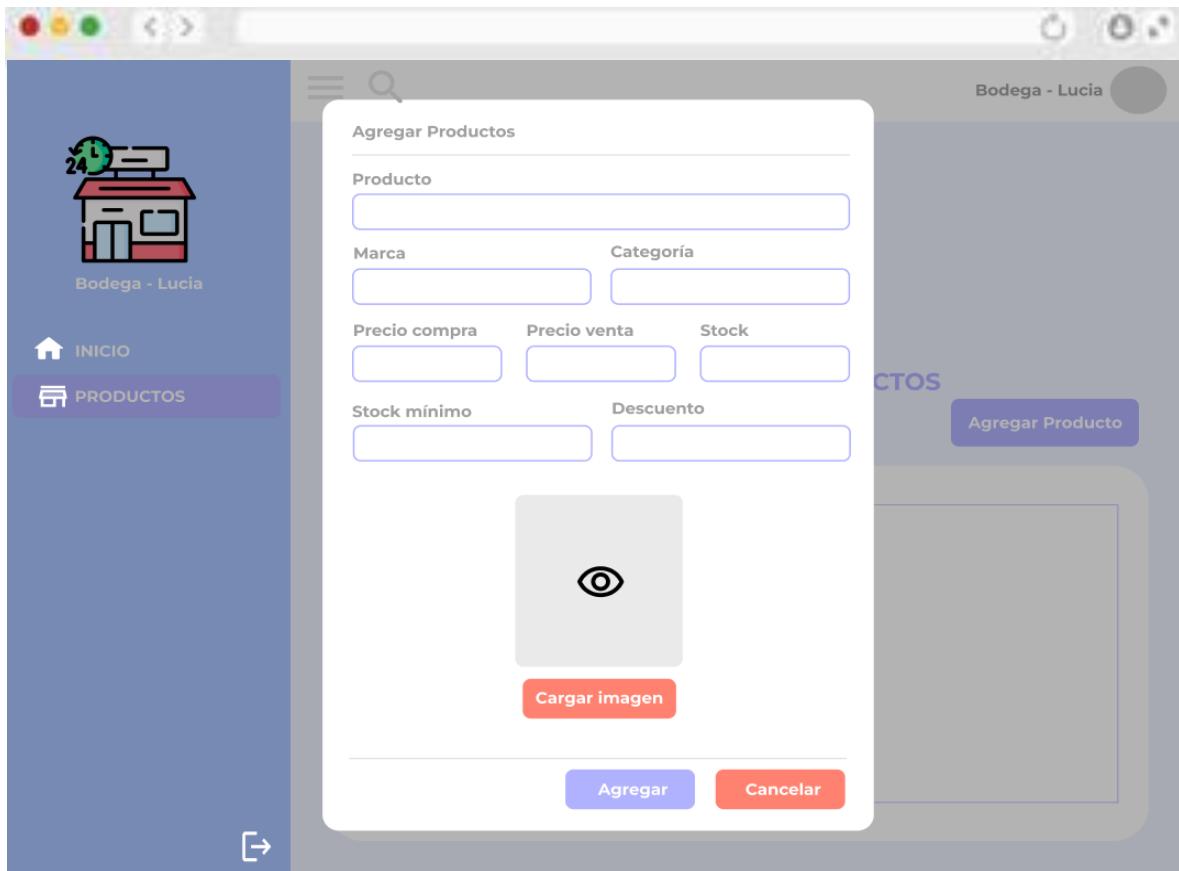


Figura 28. Dashboard para la administración de productos para las bodegas (Modelo 2)

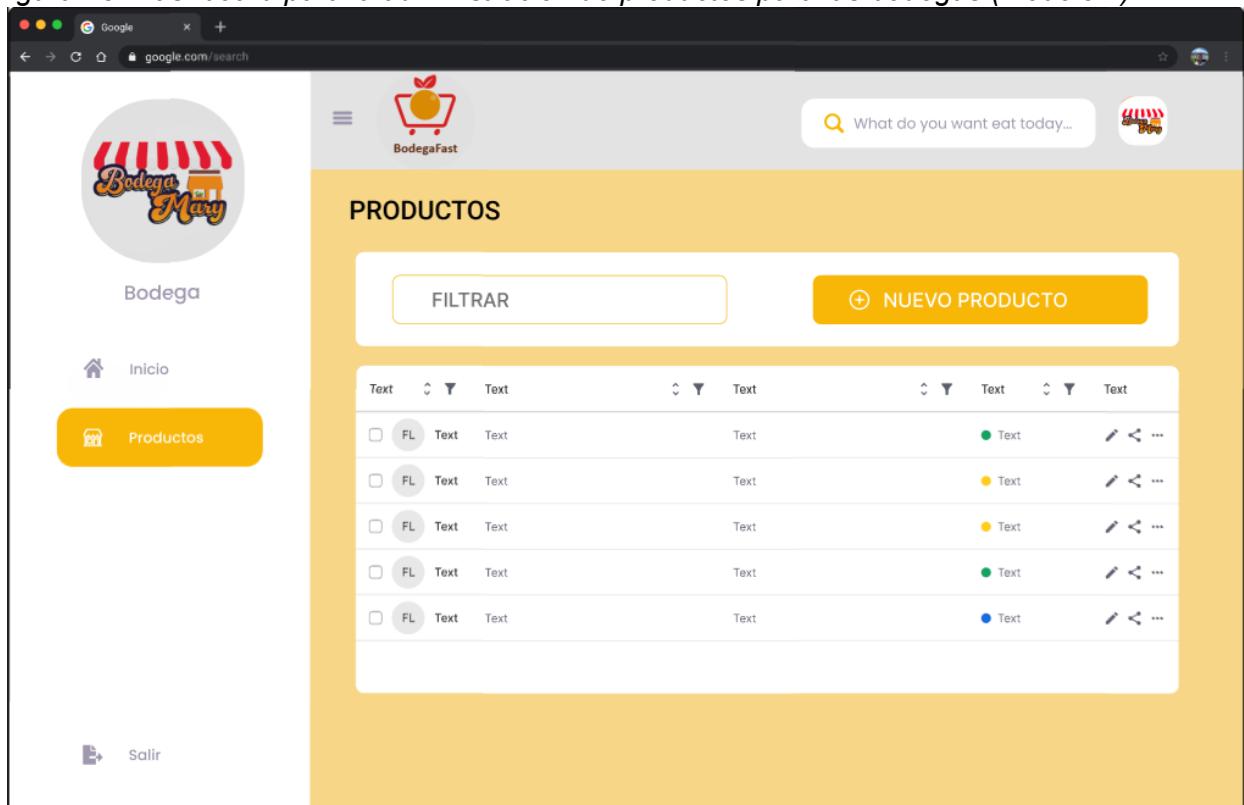


Figura 29. Dashboard para la administración de productos para las bodegas (Modelo 2 modal)

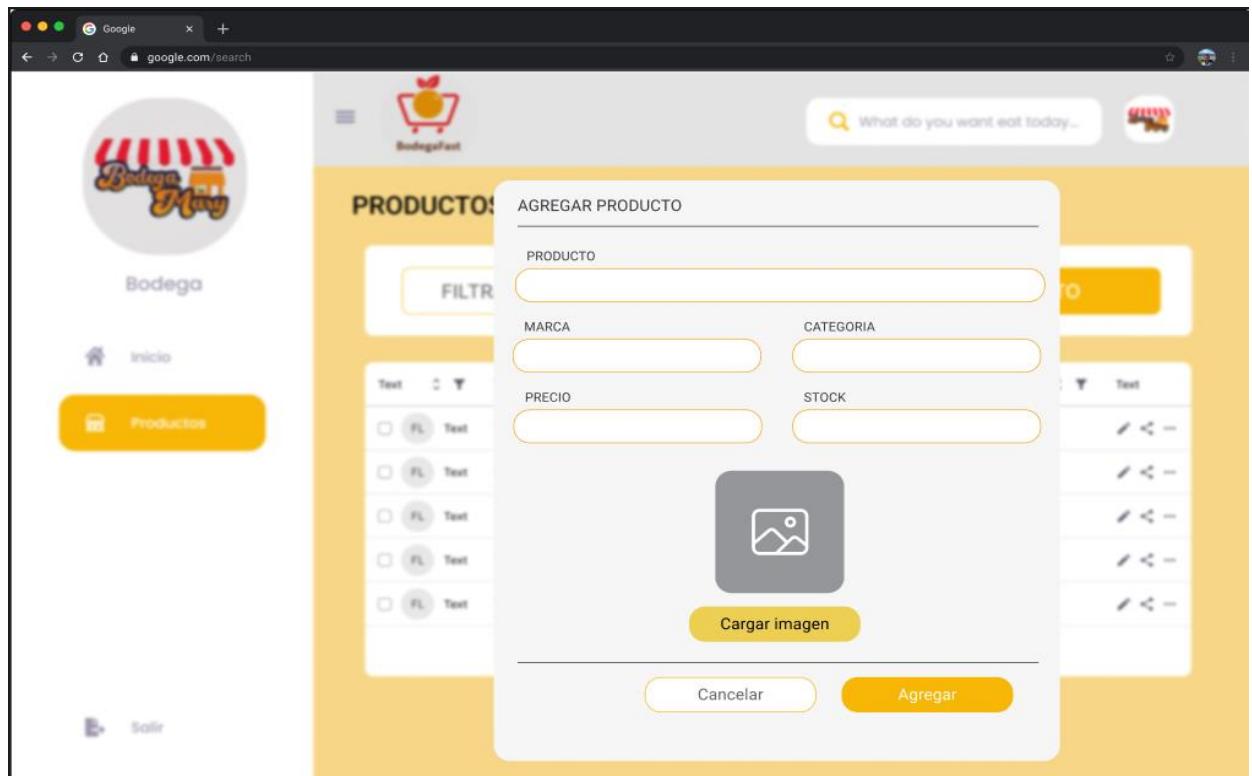


Figura 30. Dashboard para la administración de productos para las bodegas (Modelo 3)



Figura 31. Página de inicio del comercio electrónico (Modelo 1)

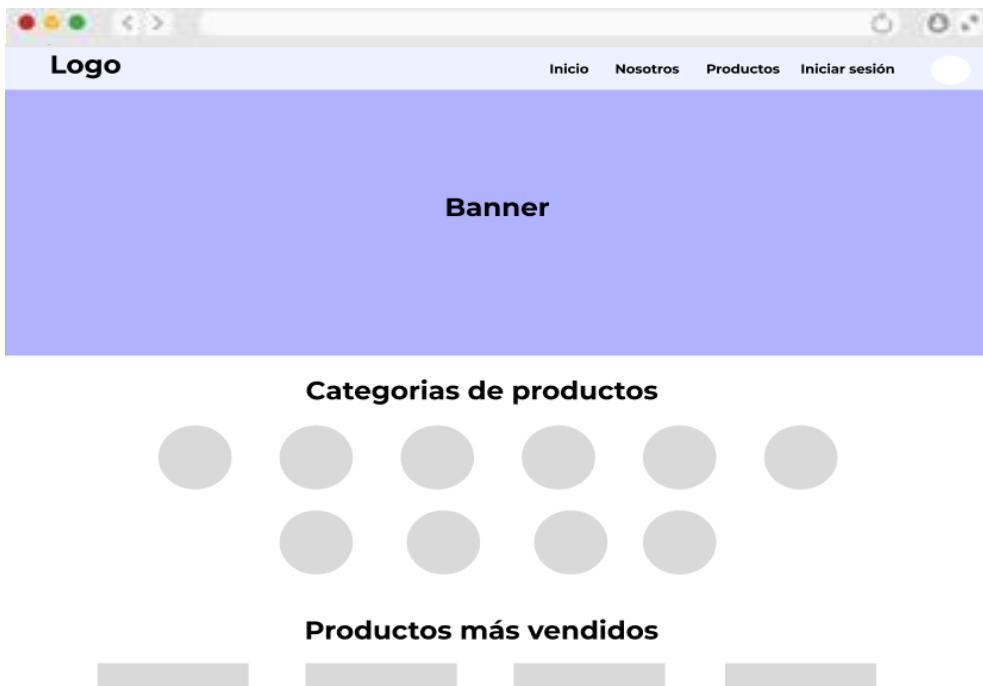


Figura 32. Página de inicio del comercio electrónico (Modelo 2)

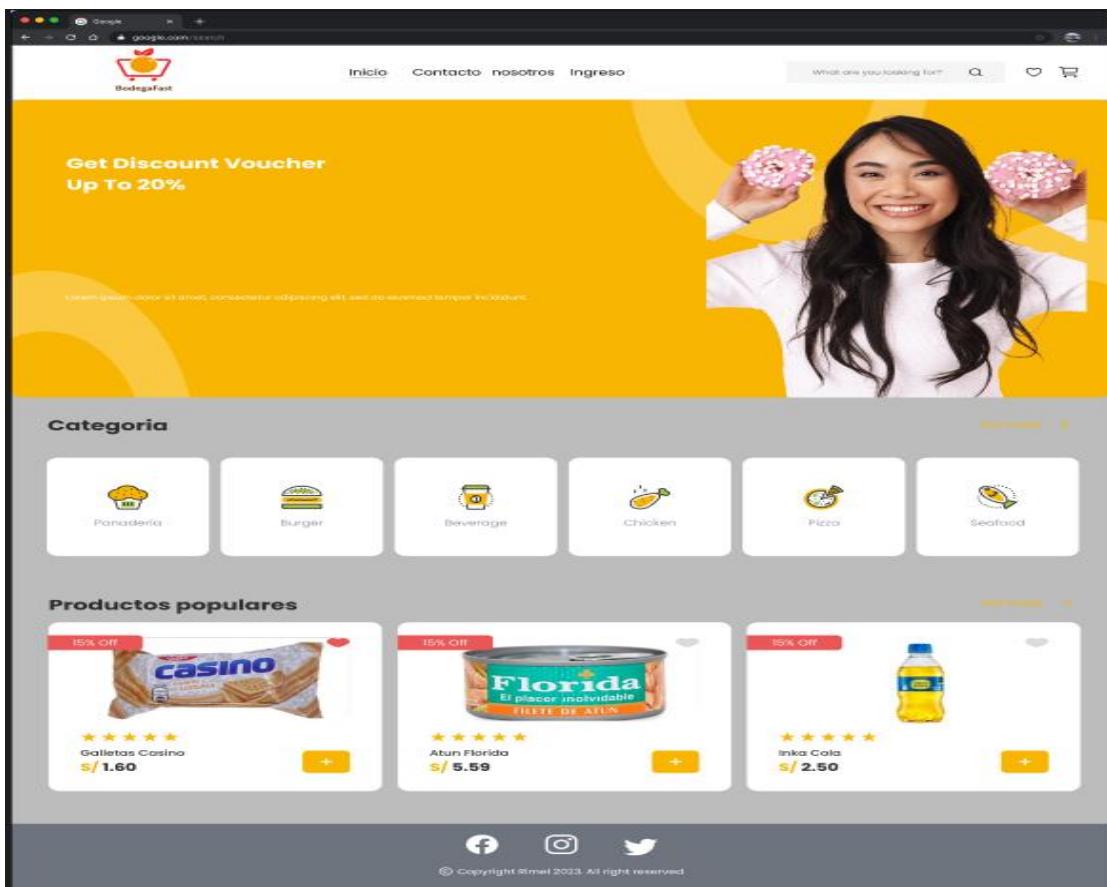


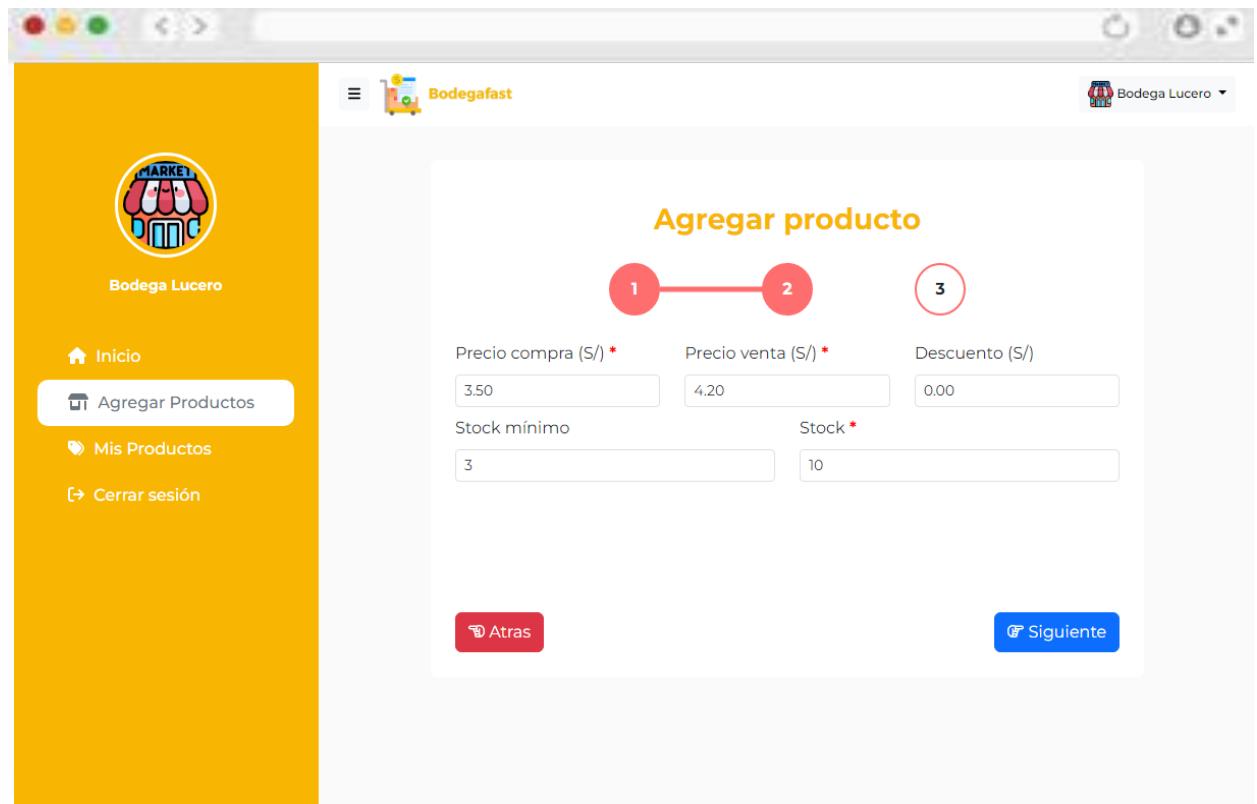
Figura 33. Página de inicio del comercio electrónico (Modelo 3)



3.8. Mockups (Facilidades de Uso) – 3 pasos – Primer entregable

Figura 34. Agregar producto (Primer paso)

Figura 35. Agregar producto (Segundo paso)



Agregar producto

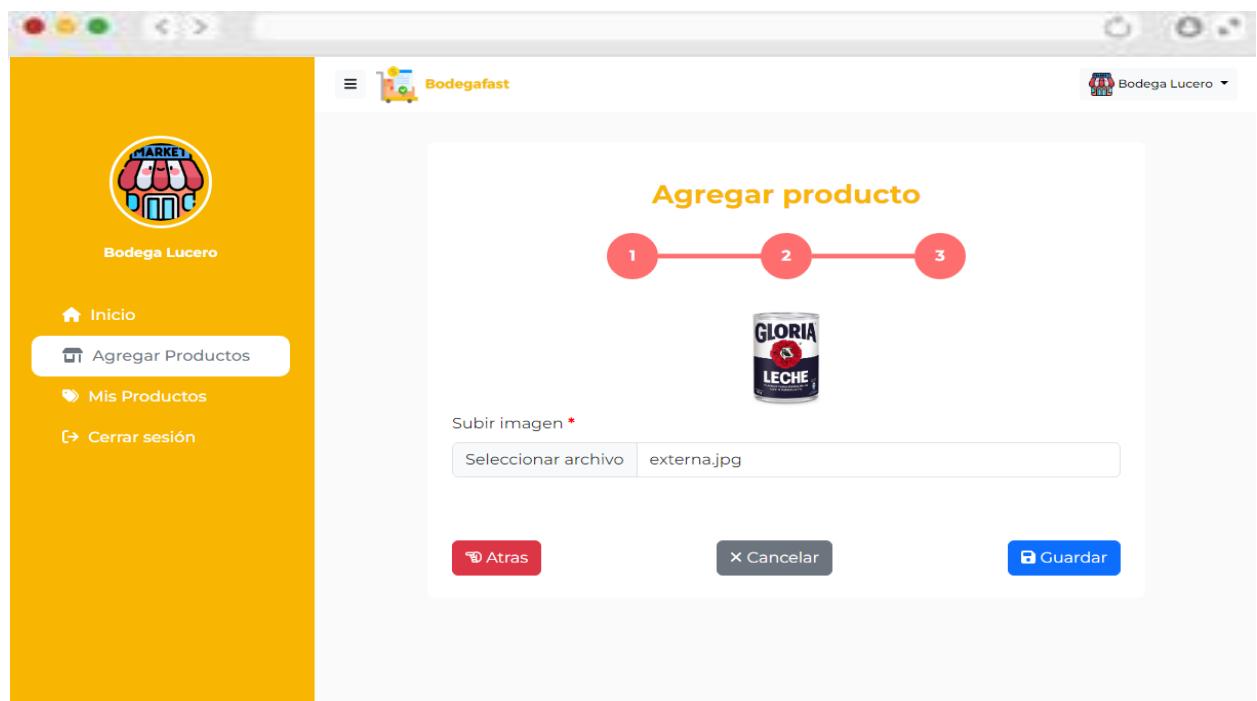
- 1
- 2
- 3

Precio compra (\$/)*	Precio venta (\$/)*	Descuento (\$/)
3.50	4.20	0.00

Stock mínimo	Stock*
3	10

[Atras](#) [Siguiente](#)

Figura 36. Agregar producto (Tercer paso)



Agregar producto

- 1
- 2
- 3



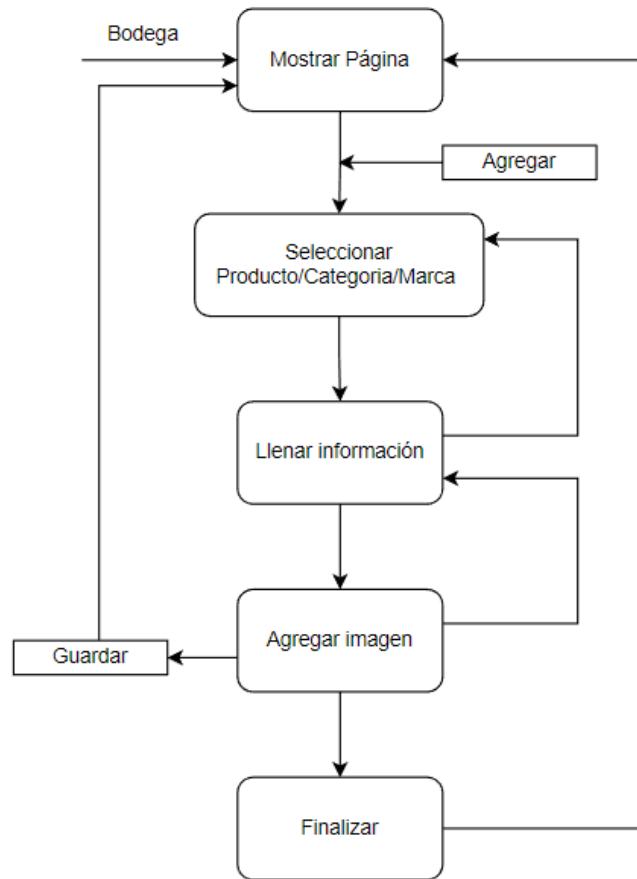
Subir imagen*

Seleccionar archivo: externajpg

[Atras](#) [Cancelar](#) [Guardar](#)

3.9. Flujo de navegación de pantallas

Figura 37. Flujo de navegación de pantalla de BodegaFast



Casos de prueba (Añadir productos)

1. M-S-LL-A-F
2. M-S-LL-S-LL-A-F
3. M-S-LL-A-LL-A-F
4. M-S-LL-S-LL-A-LL-A-F
5. M-S-LL-A-G-M
6. M-S-LL-S-LL-A-G-M
7. M-S-LL-S-LL-A-LL-A-G-M

Leyenda:

M: Mostrar pagina

S: Seleccionar categoría y marca

LL: Llenar la información del stock y costos

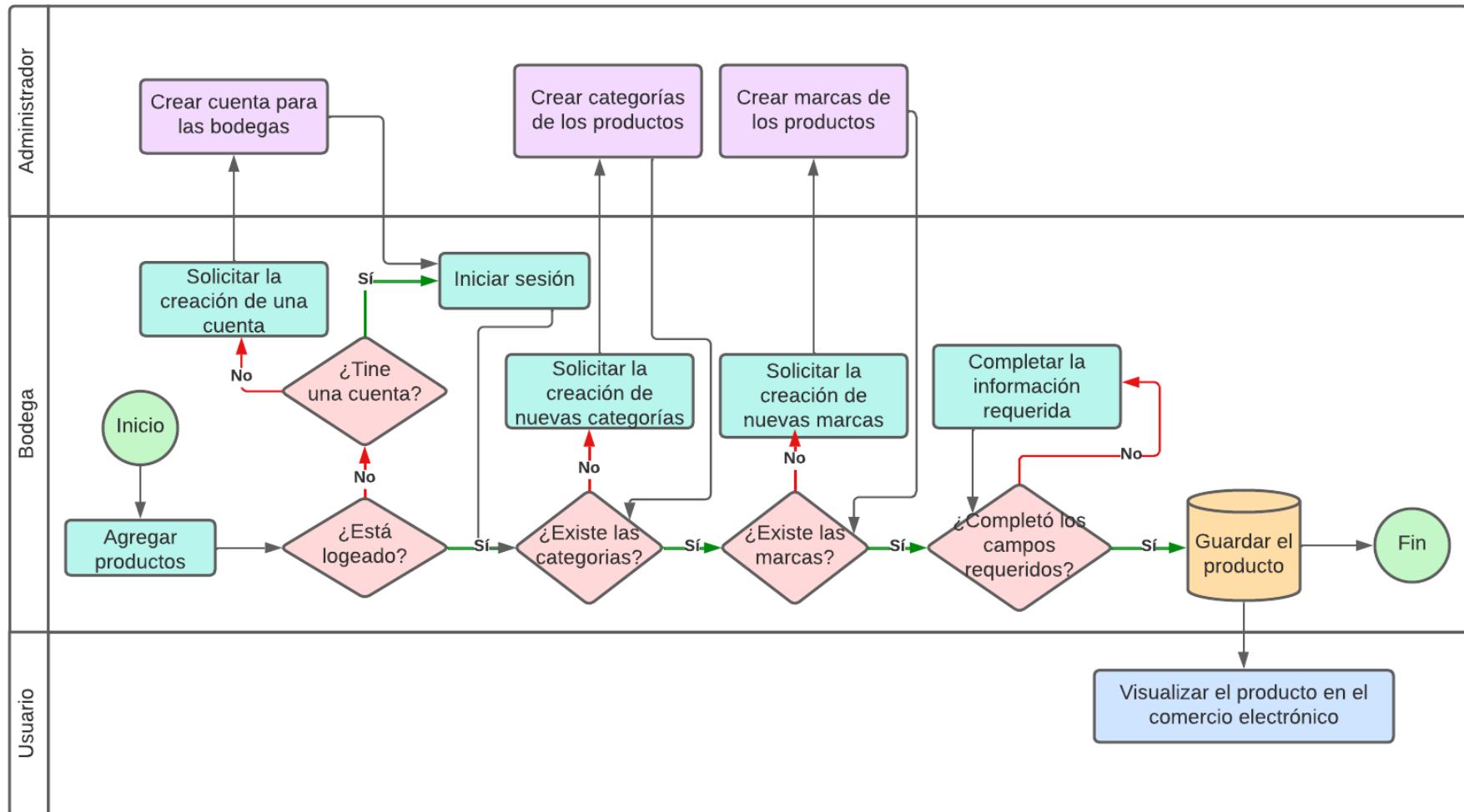
A: Agregar Imagen

F: Finalizar

G: Guardar

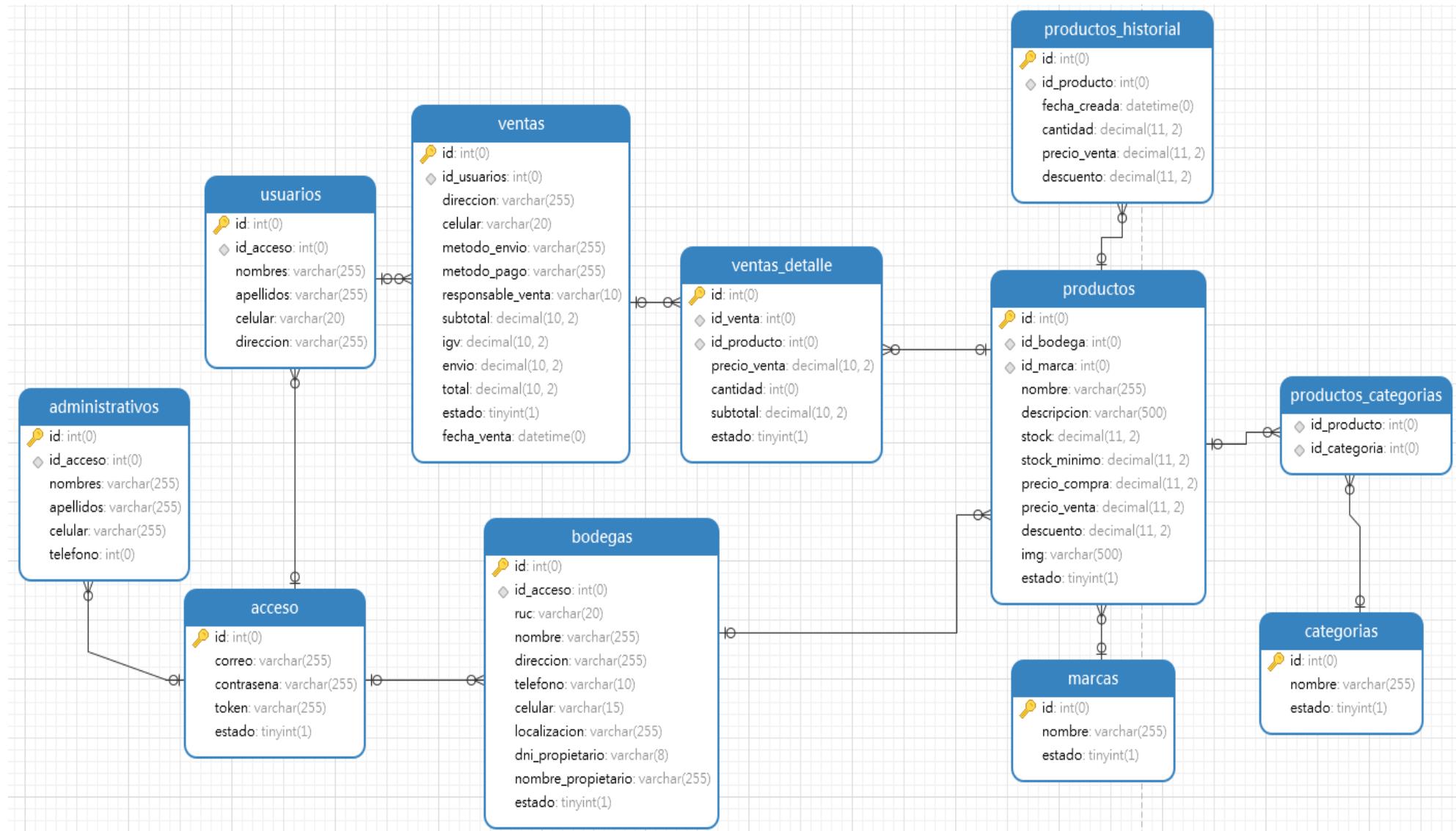
3.10. Diagrama de procesos

Figura 38. Diagrama de procesos de BodegaFast



3.11. Diagrama entidad relación física

Figura 39. Diagrama de entidad relación física de BodegaFast



Script generado del diagrama entidad relación en formato SQL

Figura 40. Script del diagrama entidad relaci髇 de BodegaFast

```

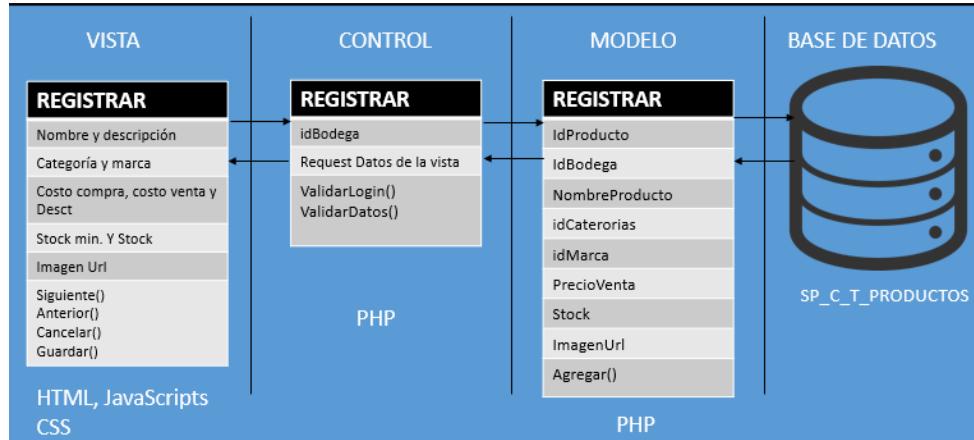
1 • CREATE DATABASE bodegafast;
2 • CREATE TABLE `marcas` (
3     `id` int(11) NOT NULL AUTO_INCREMENT,
4     `nombre` varchar(255) NULL,
5     `estado` tinyint(1) NULL,
6     PRIMARY KEY (`id`)
7 );
8 • CREATE TABLE `productos` (
9     `id` int(11) NOT NULL AUTO_INCREMENT,
10    `id_bodega` int(11) NULL,
11    `id_marca` int(11) NULL,
12    `nombre` varchar(255) NULL,
13    `descripcion` varchar(500) NULL,
14    `stock` decimal(11,2) NULL,
15    `stock_minimo` decimal(11,2) NULL,
16    `precio_compra` decimal(11,2) NULL,
17    `precio_venta` decimal(11,2) NULL,
18    `descuento` decimal(11,2) NULL,
19    `img` varchar(255) NULL,
20    `estado` tinyint(1) NULL,
21    PRIMARY KEY (`id`)
22 );
23 • CREATE TABLE `categorias` (
24     `id` int(11) NOT NULL AUTO_INCREMENT,
25     `nombre` varchar(255) NULL,
26     `estado` tinyint(1) NULL,
27     PRIMARY KEY (`id`)
28 );
29 • CREATE TABLE `productos_categorias` (
30     `id_producto` int(11) NULL,
31     `id_categoria` int(11) NULL
32 );
33 • CREATE TABLE `bodegas` (
34     `id` int(11) NOT NULL AUTO_INCREMENT,
35     `id_acceso` int(11) NULL,
36     `ruc` varchar(11) NULL,
37     `nombre` varchar(255) NULL,
38     `direccion` varchar(255) NULL,
39     `telefono` varchar(7) NULL,
40     `celular` varchar(9) NULL,
41     `localizacion` varchar(255) NULL,
42     `dni_propietario` varchar(8) NULL,
43     `nombre_propietario` varchar(255) NULL,
44     PRIMARY KEY (`id`)
45 );
46 • CREATE TABLE `accesos` (
47     `id` int(11) NOT NULL AUTO_INCREMENT,
48     `correo` varchar(255) NULL,
49     `contrasena` varchar(255) NULL,
50     `token` varchar(255) NULL,
51     `estado` tinyint(1) NULL,
52     PRIMARY KEY (`id`),
53     UNIQUE INDEX (`correo` ASC)
54 );
55 • CREATE TABLE `administrativos` (
56     `id` int(11) NOT NULL AUTO_INCREMENT,
57     `id_acceso` int(11) NULL,
58     `nombres` varchar(255) NULL,
59     `apellidos` varchar(255) NULL,
60     `celular` varchar(9) NULL,
61     `telefono` varchar(7) NULL,
62     PRIMARY KEY (`id`)
63 );
76 • ALTER TABLE `productos` ADD FOREIGN KEY (`id_marca`) REFERENCES `marcas` (`id`);
77 • ALTER TABLE `productos_categorias` ADD FOREIGN KEY (`id_producto`) REFERENCES `productos` (`id`);
78 • ALTER TABLE `productos_categorias` ADD FOREIGN KEY (`id_categoria`) REFERENCES `categorias` (`id`);
79 • ALTER TABLE `productos` ADD FOREIGN KEY (`id_bodega`) REFERENCES `bodegas` (`id`);
80 • ALTER TABLE `bodegas` ADD FOREIGN KEY (`id_acceso`) REFERENCES `accesos` (`id`);
81 • ALTER TABLE `administrativos` ADD FOREIGN KEY (`id_acceso`) REFERENCES `accesos` (`id`);
82 • ALTER TABLE `usuarios` ADD FOREIGN KEY (`id_acceso`) REFERENCES `accesos` (`id`);

```

3.12. Diagrama de clases de software

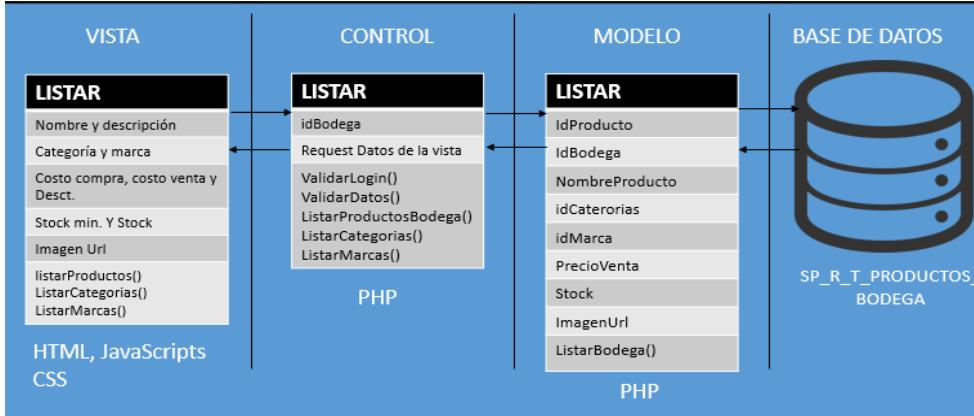
Registrar productos

Figura 41. Registro de productos de BodegaFast



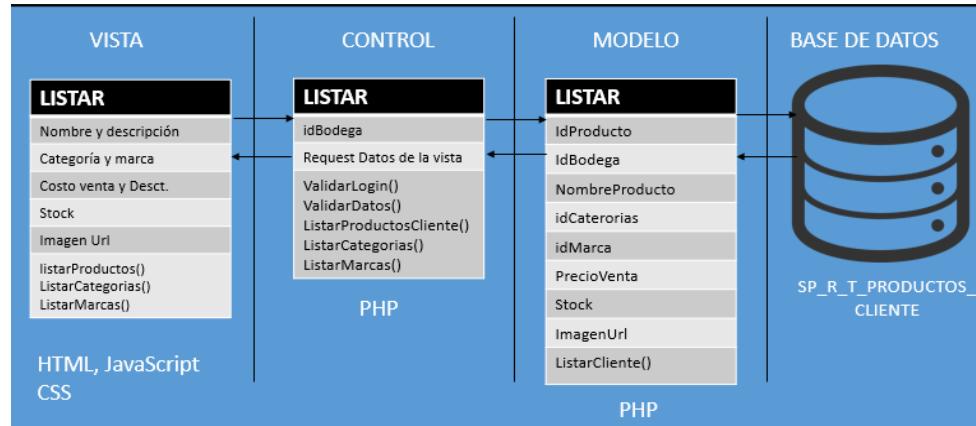
Mostrar productos para bodega

Figura 42. Mostrar productos de BodegaFast



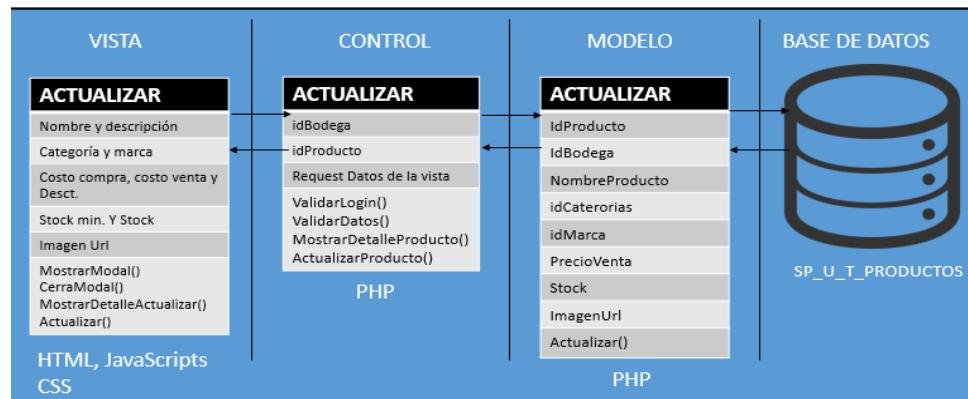
Mostrar productos para clientes

Figura 43. Mostrar productos de clientes de BodegaFast



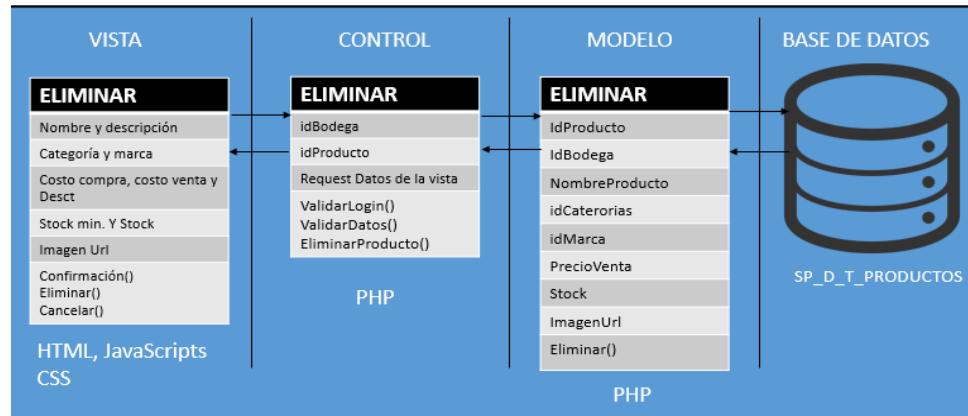
Actualizar productos

Figura 44. Actualizar productos de BodegaFast



Eliminar productos

Figura 45. Eliminar productos de BodegaFast



3.13. Vistas desarrolladas – primer entregable

Figura 46. Vista principal

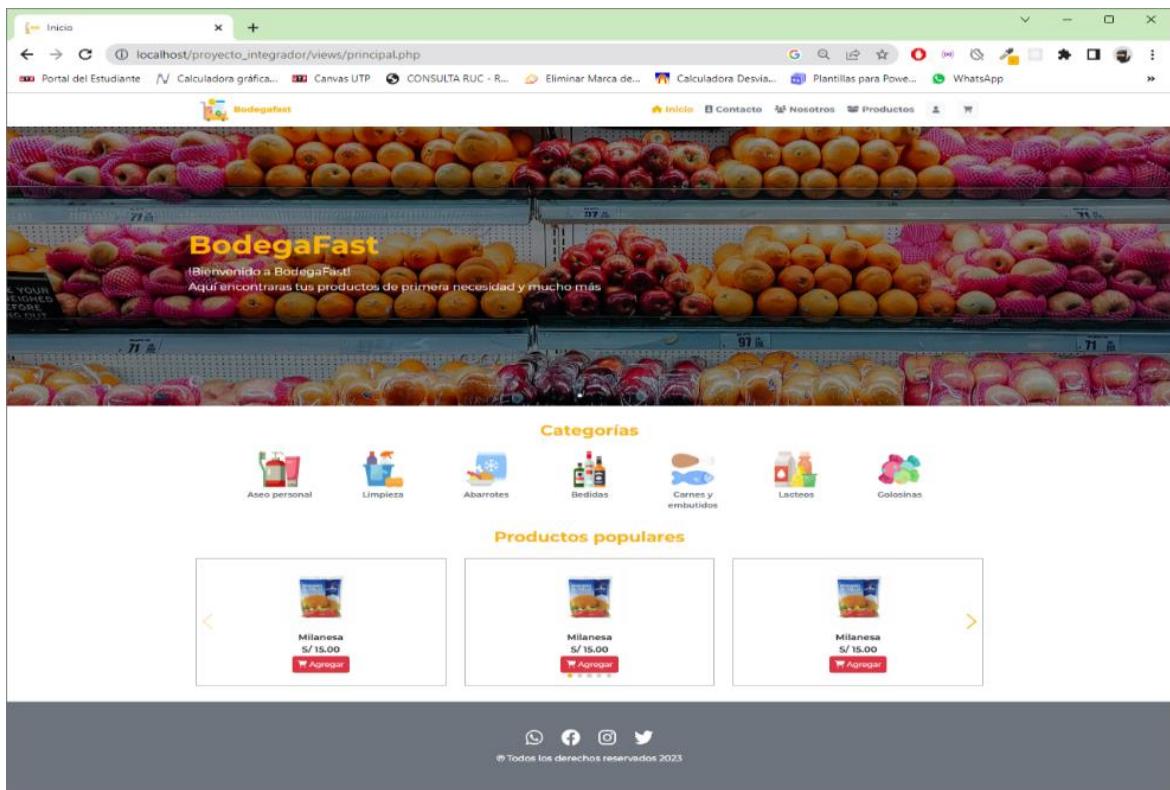


Figura 47. Vista iniciar sesión y registro

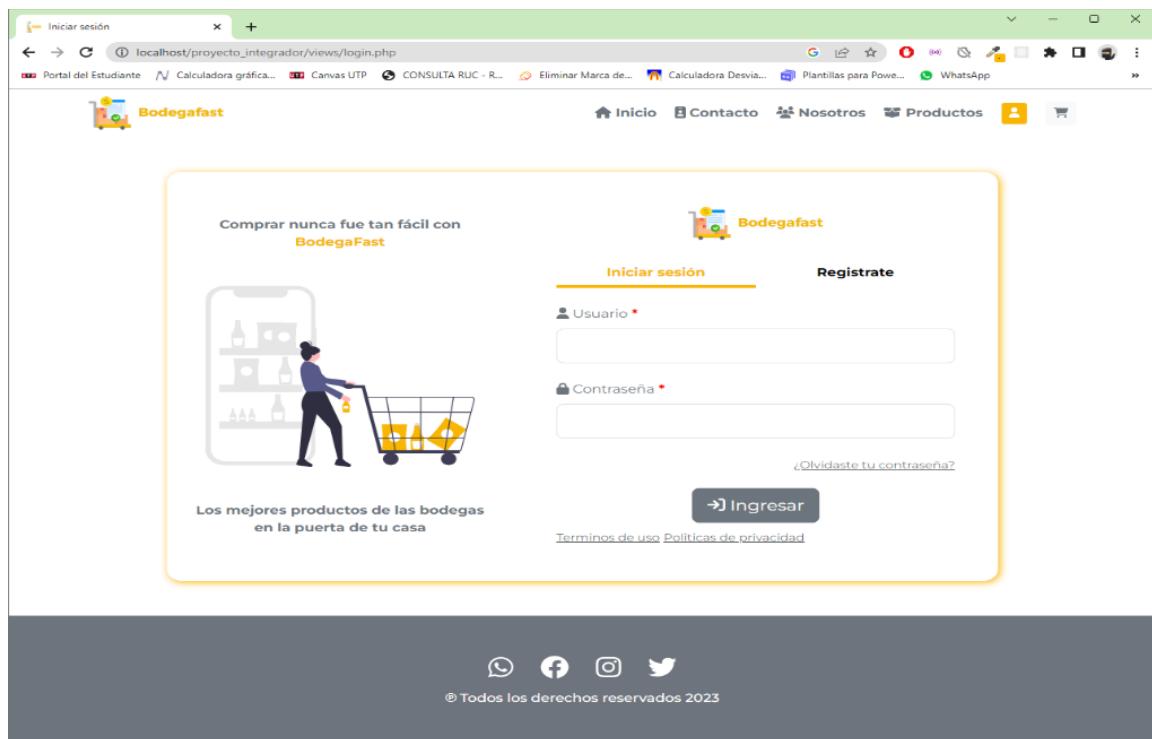


Figura 48. Vista agregar bodega

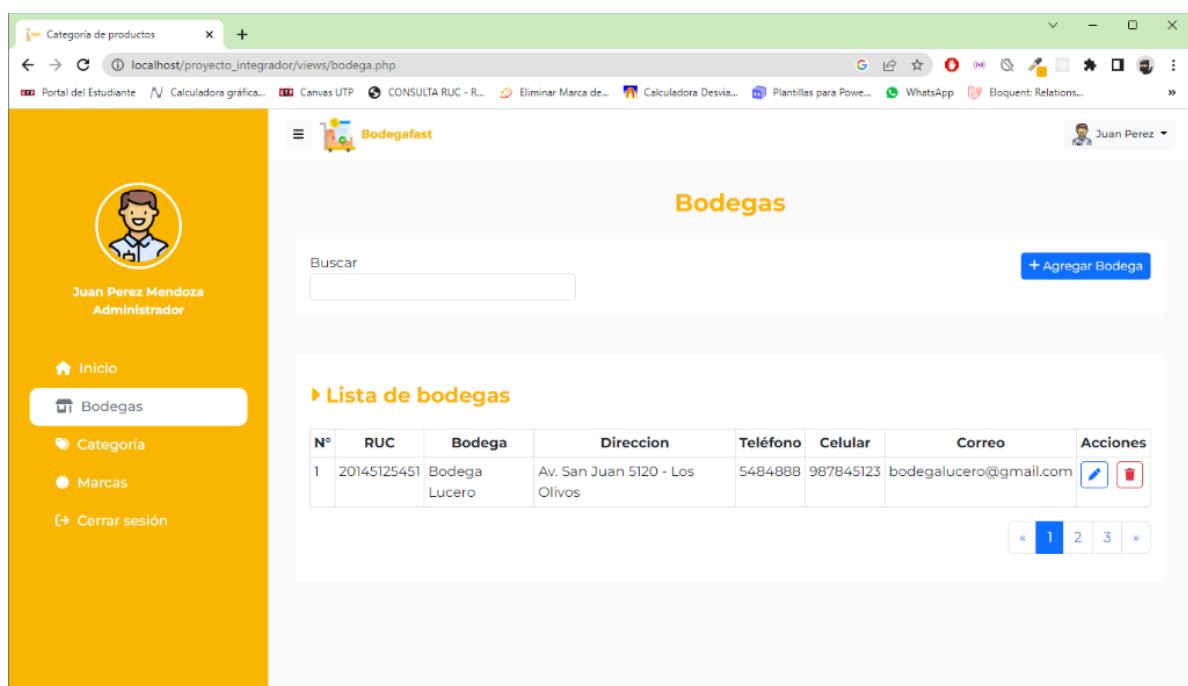


Figura 49. Vista agregar categorías

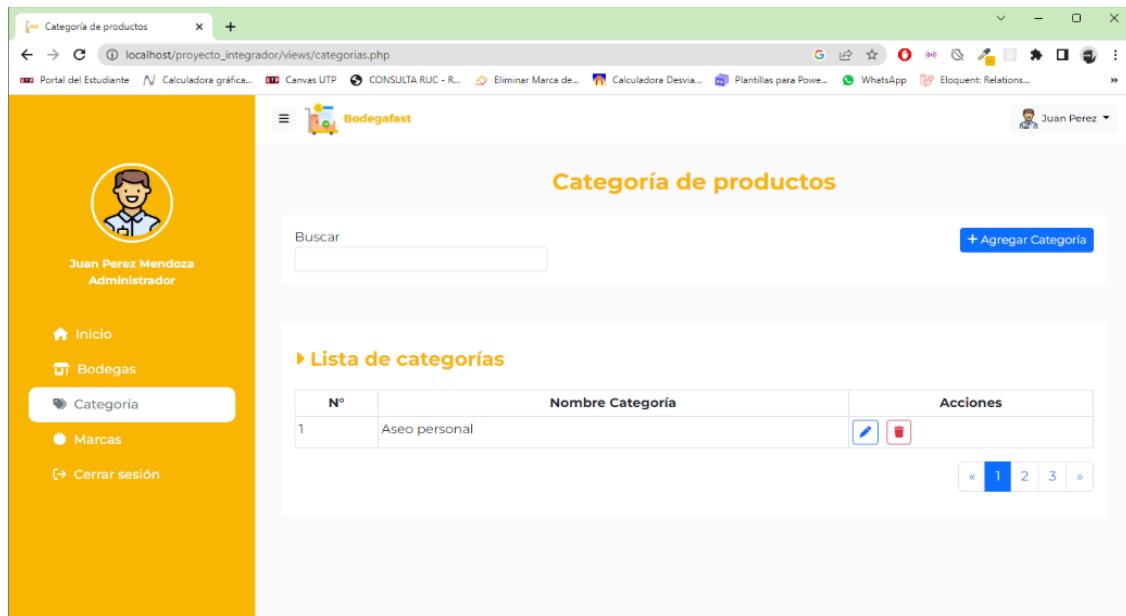


Figura 50. Vista agregar marcas

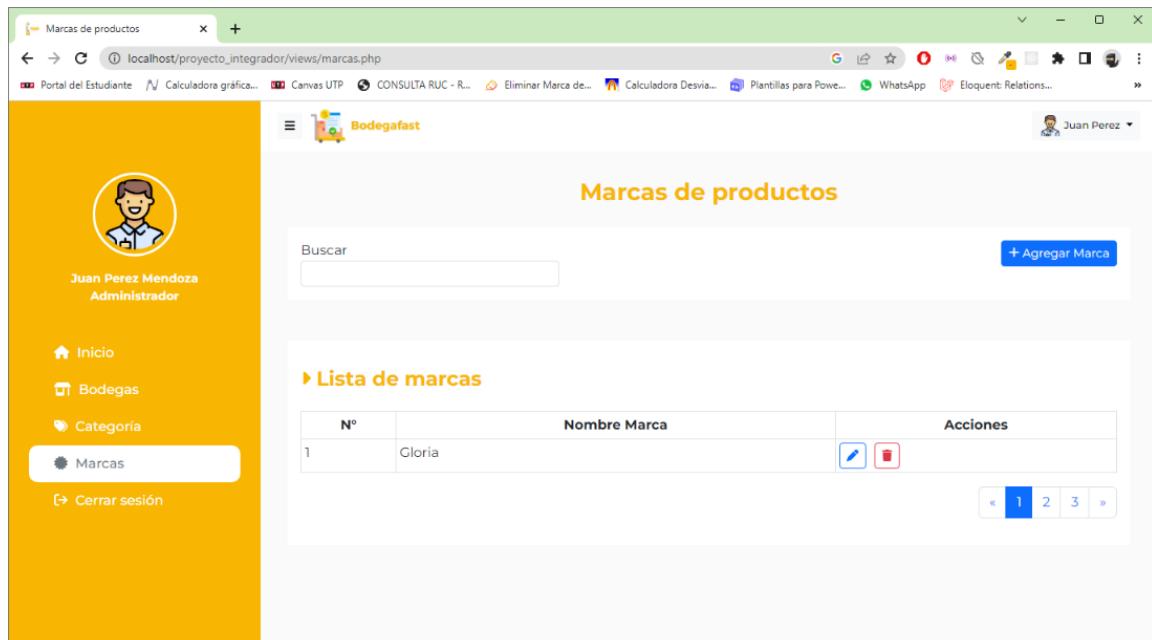


Figura 51. Vista agregar productos

Agregar producto

- 1
- 2
- 3

Producto *

Descripción

Categorías *

Marca *

Siguiente

Figura 52. Vista mis productos

Mis productos

Lista de productos

Nº	Producto	Descripción	Categorías	Marca	Precio compra	Precio venta	Stock mínimo	Stock Actual	Acciones
1	Leche Gloria Azul	Lata de 100ml grande	Lacteos	Gloria	S/ 3.50	S/ 4.20	2	10	

« 1 2 3 »

3.14. Mockups (Facilidades de Uso) – 3 pasos – Segundo entregable

Figura 53. Carrito de Compras (Primer paso)

The screenshot shows a web browser window with three tabs: 'Carrito de compras' (selected), 'Carrito de compras' (unselected), and 'Rúbrica de Avance de Proyecto' (unselected). The main content area is titled 'MI CARRITO DE COMPRAS'. It displays a table of selected products:

Nº	Bodega	Producto	Precio	Cantidad	Subtotal	Acciones
1	BODEGA LUCERO	Café Instantáneo Nescafé Tradición Frasco 170g	S/ 15.00	2	S/ 30.00	
2	BODEGA LUCERO	Leche Gloria Azul de 170g	S/ 2.20	1	S/ 2.20	
3	BODEGA LUCERO	Leche Gloria Azul de 400g	S/ 4.20	1	S/ 4.20	

The total amount is displayed as 'Total: S/ 36.40'. Below the table is a blue button labeled 'Siguiente' (Next). At the bottom of the page is a dark grey footer bar with social media icons for WhatsApp, Facebook, Instagram, and Twitter.

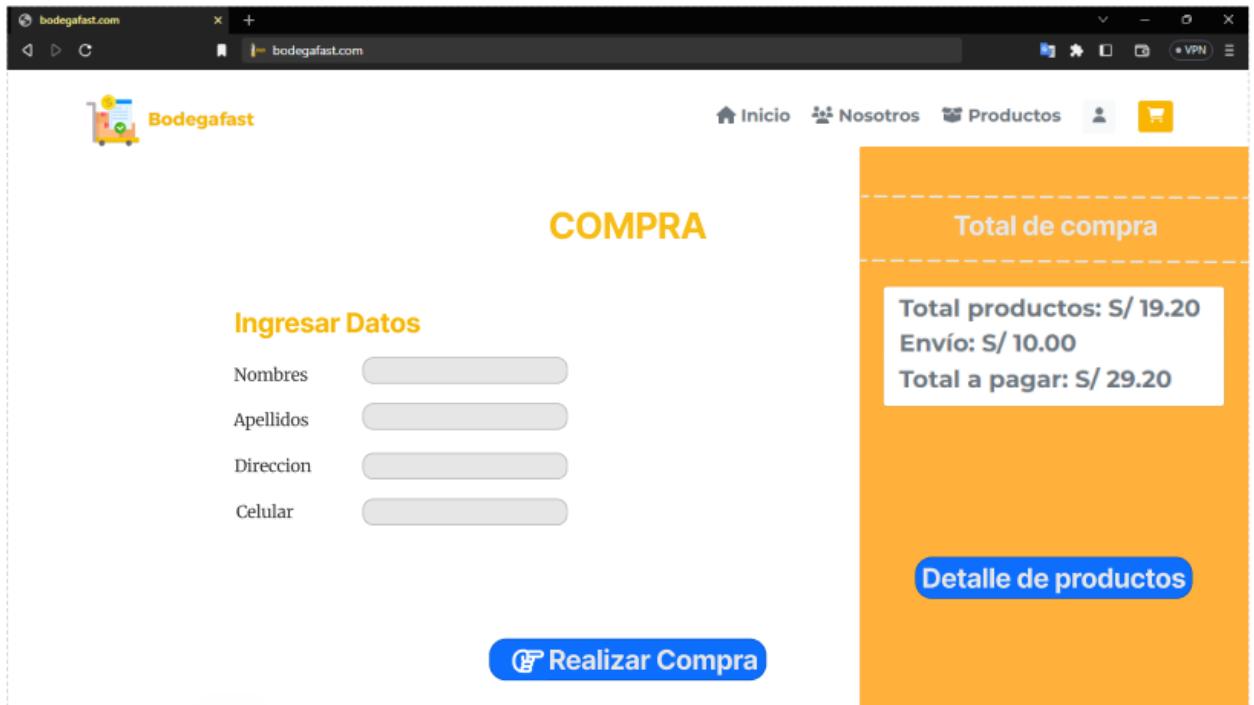
Figura 54. Carrito de Compras (Segundo paso)

The screenshot shows a web browser window with three tabs: 'Carrito de compras', 'Carrito de compras', and 'Rúbrica de Avance de Proyecto'. The main content area is titled 'MI CARRITO DE COMPRAS' with a progress bar showing steps 1 and 2 completed. The form section is titled 'Datos personales' and contains fields for 'Nombres' (jean pier) and 'Apellidos' (carra), both with placeholder text. There is also a 'Dirección *' field containing 'Av. Los alisos 120 - Los Olivos' and a 'Celular *' field which is empty. Below the form are 'Atras' and 'Siguiente' buttons. At the bottom, there are social media sharing icons.

Figura 55. Carrito de Compras (Tercer paso)

The screenshot shows a web browser window with three tabs: 'Carrito de compras', 'Carrito de compras', and 'Rúbrica de Avance de Proyecto'. The main content area is titled 'MI CARRITO DE COMPRAS' with a progress bar showing steps 1, 2, and 3 completed. The section is titled 'Finalizar compra' and displays the total information: 'Total productos: S/ 36.40', 'Envío: S/ 10.00', and 'Total a pagar: S/ 46.40'. A radio button is selected for 'Pago contra entrega'. Below the information are 'Atras', 'Cancelar', and 'Comprar' buttons. At the bottom, there are social media sharing icons.

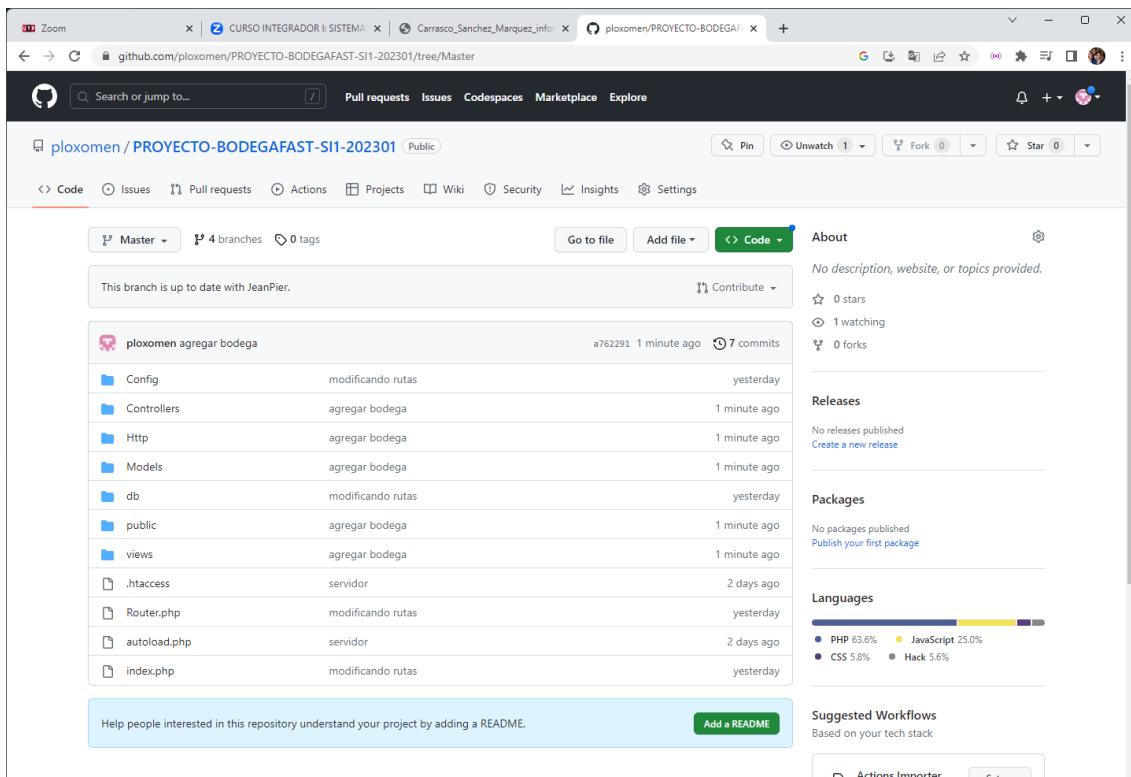
Figura 56. Carrito de Compras (Modelo 2)



3.15. GitHub carpetas y entornos de desarrollo

Proyecto “[PROYECTO-BODEGAFAST-SI1-202301](#)” almacenado en GitHub

Figura 57. GitHub carpeta y entornos de desarrollo



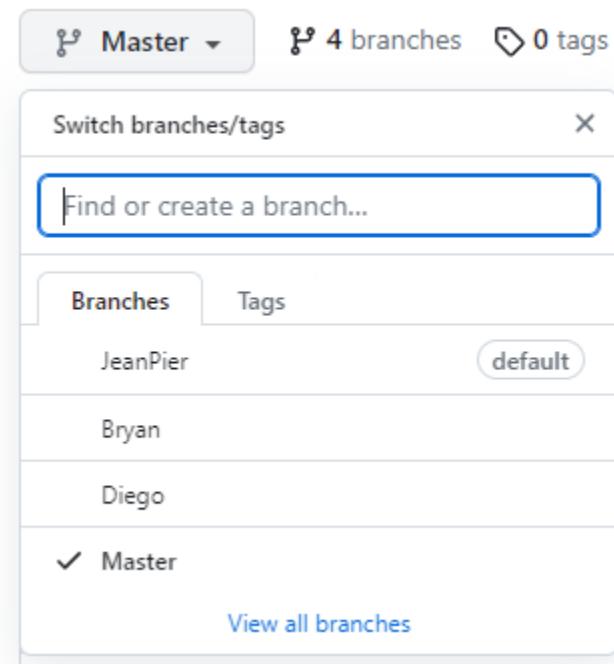
Enlace <https://github.com/ploxomen/PROYECTO-BODEGAFAST-SI1-202301.git>

Figura 58. Estructura de carpetas

Config	modificando rutas	yesterday
Controllers	agregar bodega	1 minute ago
Http	agregar bodega	1 minute ago
Models	agregar bodega	1 minute ago
db	modificando rutas	yesterday
public	agregar bodega	1 minute ago
views	agregar bodega	1 minute ago
.htaccess	servidor	2 days ago
Router.php	modificando rutas	yesterday
autoload.php	servidor	2 days ago
index.php	modificando rutas	yesterday

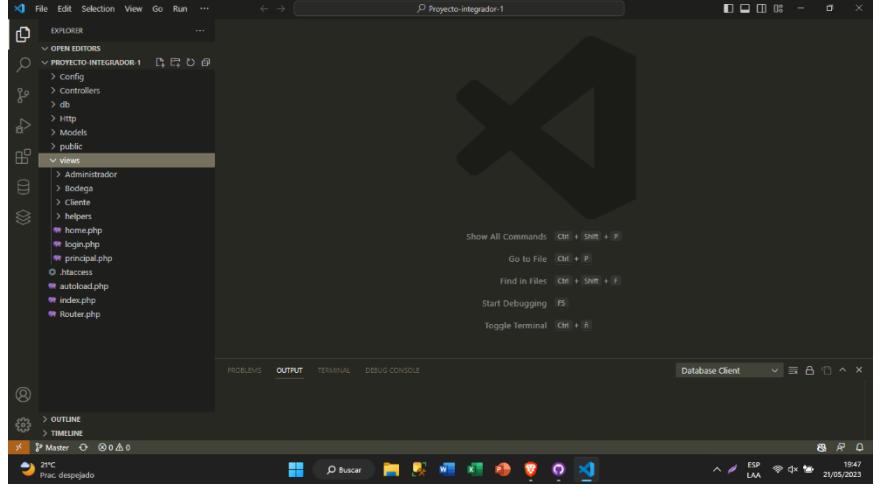
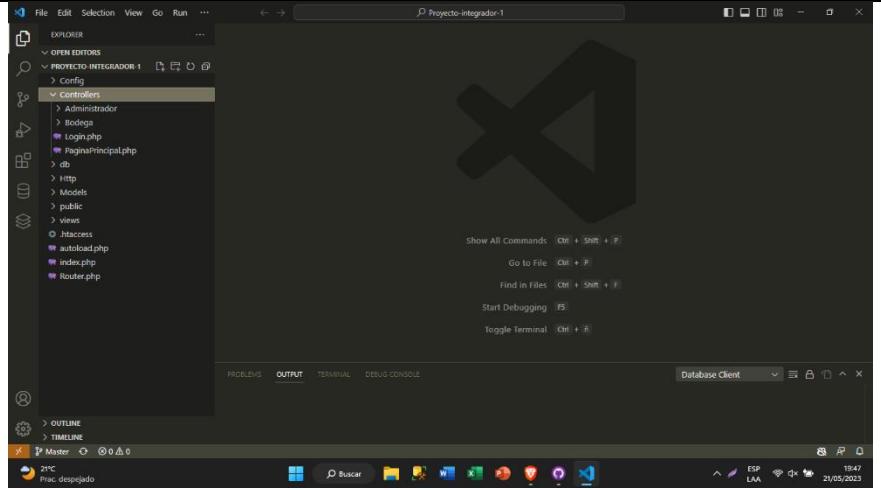
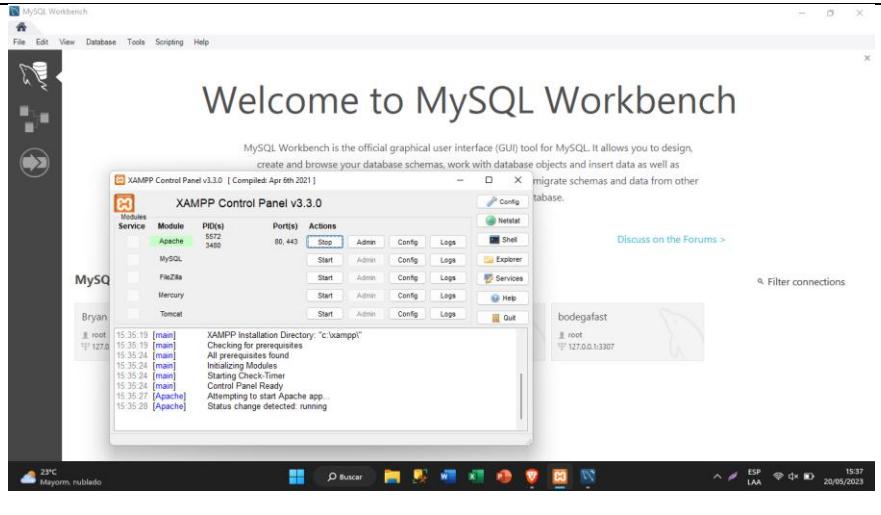
Se está trabajando en ramas, 3 ramas para los desarrolladores y una rama master que une el código de todos los programadores.

Figura 59. Ramas de desarrolladores y rama Master



3.16. Arquitectura en capas de Bryan

Navegador	
Front/Back	

Front																															
Back																															
Base de Datos	 <p>Welcome to MySQL Workbench</p> <p>XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]</p> <table border="1"> <thead> <tr> <th>Module</th> <th>Service</th> <th>Module</th> <th>Port(s)</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td></td> <td>Apache</td> <td>Apache</td> <td>80, 443</td> <td>Stop Admin Config Logs Shell</td> </tr> <tr> <td></td> <td>MySQL</td> <td>MySQL</td> <td>3306</td> <td>Start Admin Config Logs Explorer Services Help</td> </tr> <tr> <td></td> <td>FileZilla</td> <td>FileZilla</td> <td></td> <td>Start Admin Config Logs</td> </tr> <tr> <td></td> <td>Mercury</td> <td>Mercury</td> <td></td> <td>Start Admin Config Logs</td> </tr> <tr> <td></td> <td>Tomcat</td> <td>Tomcat</td> <td></td> <td>Start Admin Config Logs</td> </tr> </tbody> </table> <pre> XAMPP Installation Directory: 'c:\xampp\' 15:35:19 [main] Checking for prerequisites 15:35:24 [main] All prerequisites found 15:35:24 [main] Initializing Modules 15:35:24 [main] Bryan's Apache2.4.41 (Windows) Timer 15:35:24 [main] Control Panel Ready 15:35:27 [Apache] Attempting to start Apache app... 15:35:28 [Apache] Status change detected: running </pre>	Module	Service	Module	Port(s)	Actions		Apache	Apache	80, 443	Stop Admin Config Logs Shell		MySQL	MySQL	3306	Start Admin Config Logs Explorer Services Help		FileZilla	FileZilla		Start Admin Config Logs		Mercury	Mercury		Start Admin Config Logs		Tomcat	Tomcat		Start Admin Config Logs
Module	Service	Module	Port(s)	Actions																											
	Apache	Apache	80, 443	Stop Admin Config Logs Shell																											
	MySQL	MySQL	3306	Start Admin Config Logs Explorer Services Help																											
	FileZilla	FileZilla		Start Admin Config Logs																											
	Mercury	Mercury		Start Admin Config Logs																											
	Tomcat	Tomcat		Start Admin Config Logs																											

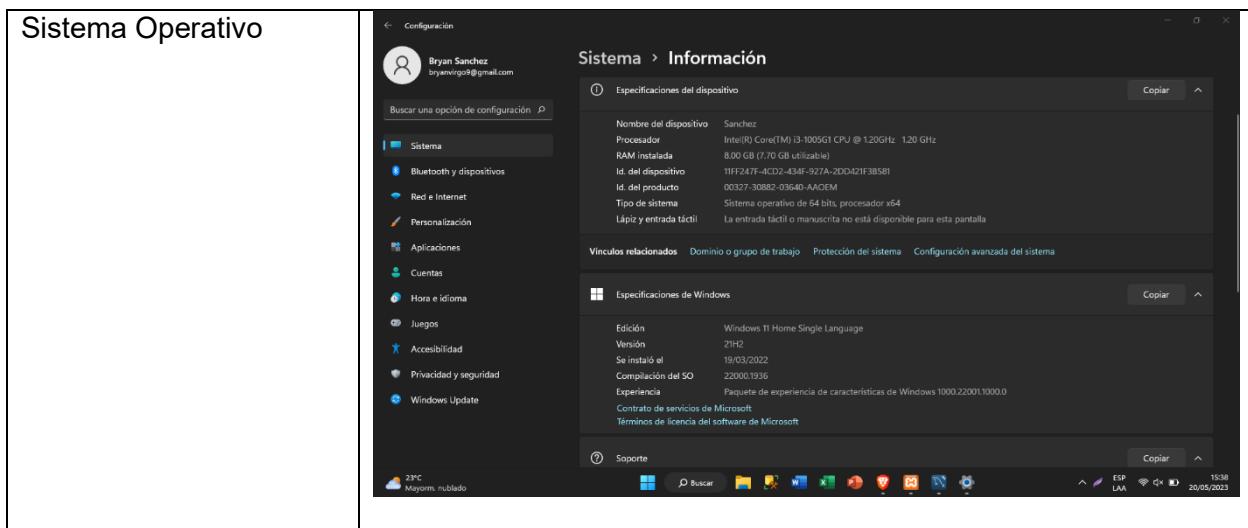
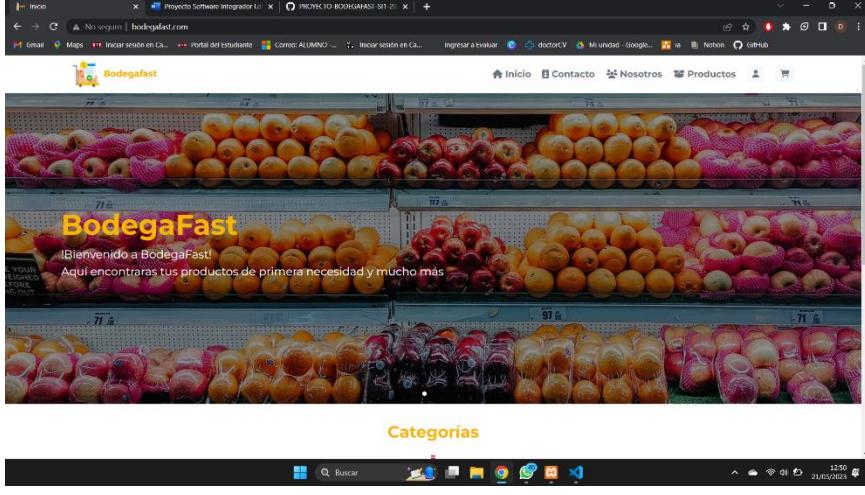
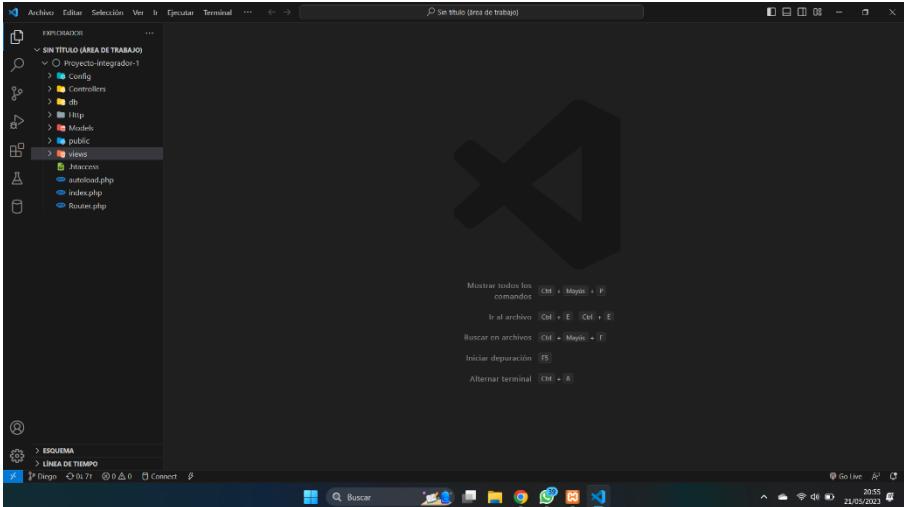
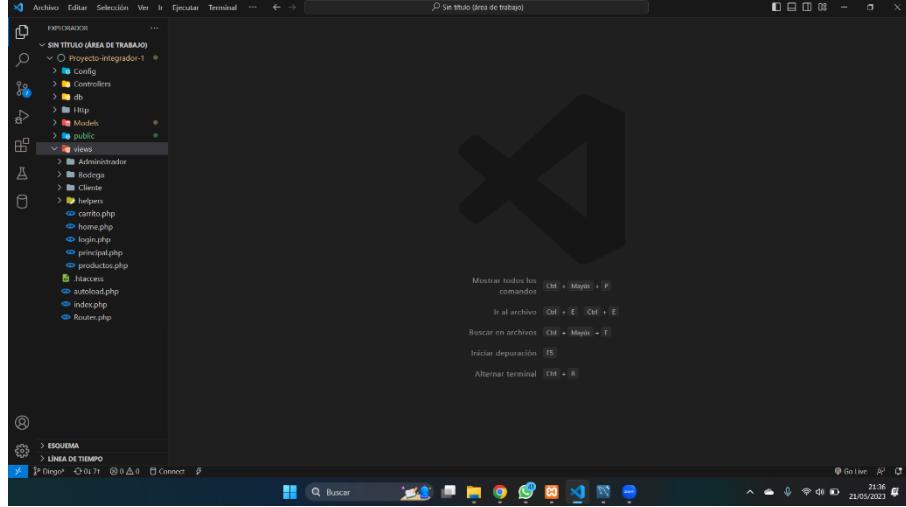
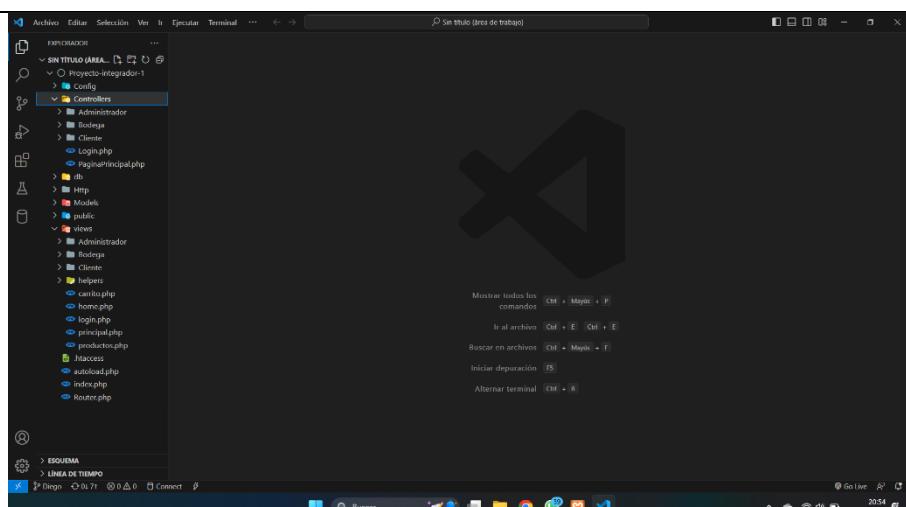
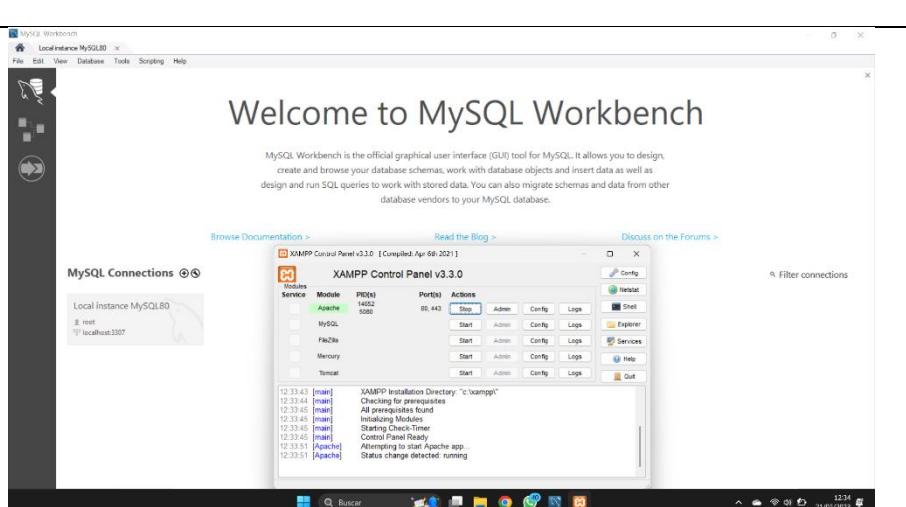


Tabla 2. Arquitectura de desarrollo de Bryan

3.17. Arquitectura en capas de Diego

Navegador	
Front/Back	

Front 	Back 
Base de Datos 	

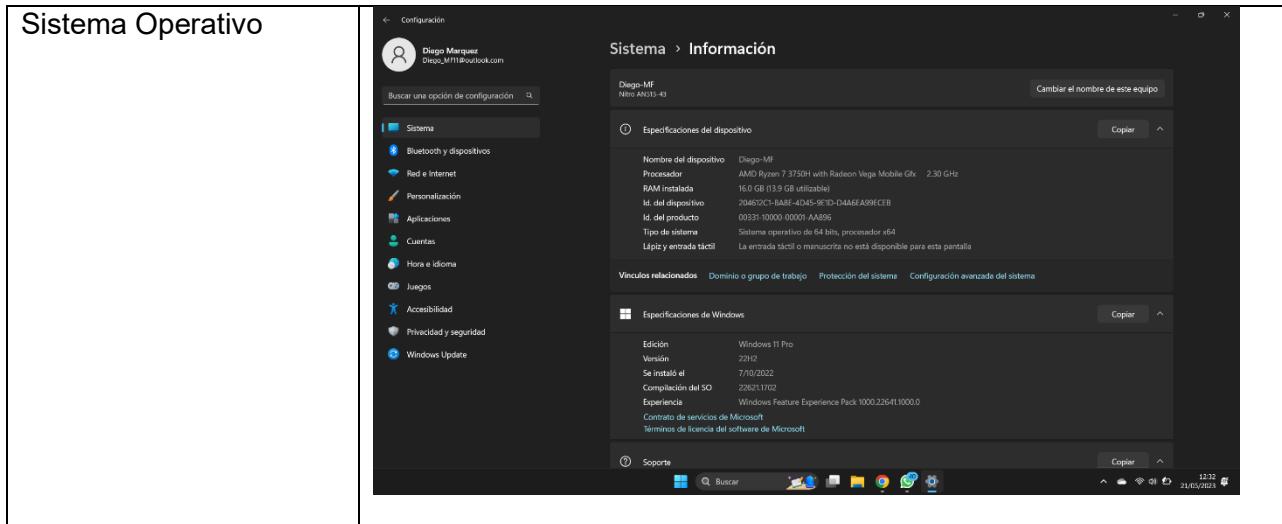
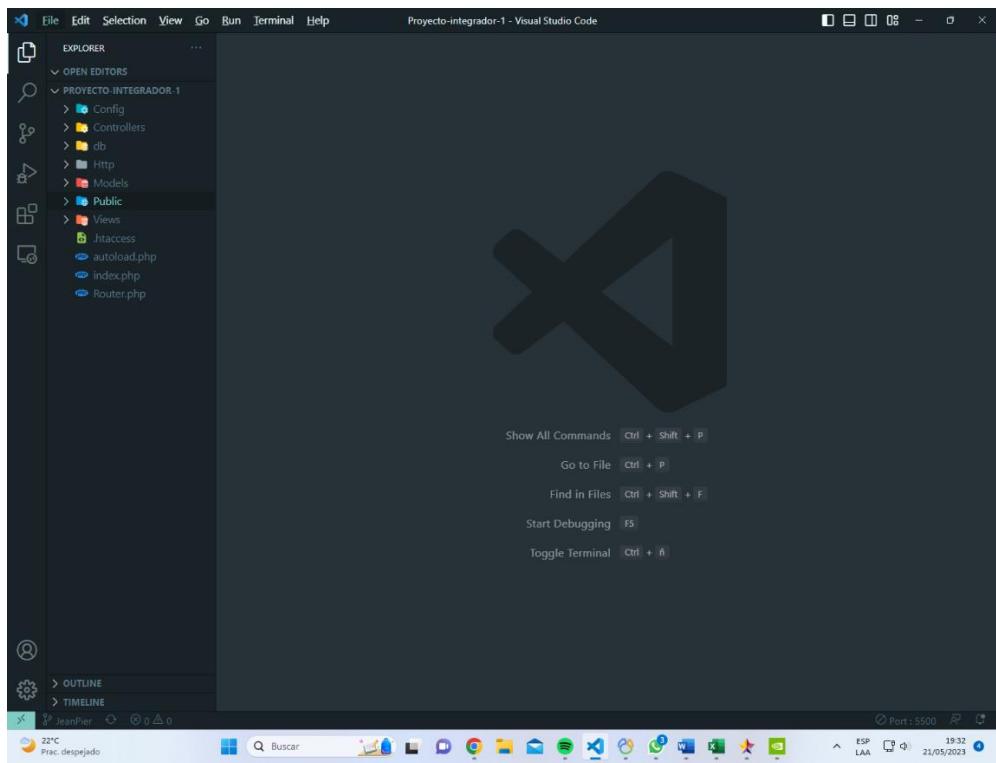


Tabla 3. Arquitectura de desarrollo Diego

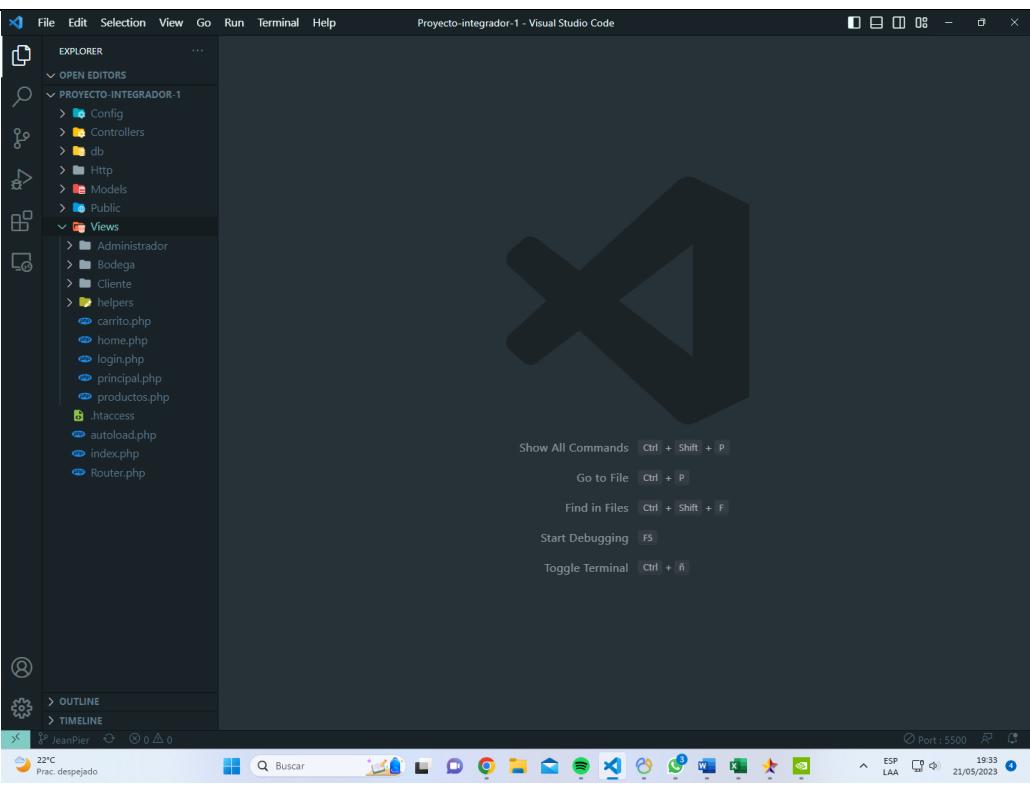
3.18. Arquitecturas en capas de Jean Pier

Navegador	
-----------	--

Front/Back



Front



<p>Back</p>	
<p>Base de Datos</p>	<pre> XAMPP Installation Directory: "c:\xampp" 10:00:35 [main] Checking for prerequisites 10:00:35 [main] All prerequisites found 10:00:36 [main] Initializing Modules 10:00:36 [main] Starting Check-Timer 10:00:36 [main] Control Panel Ready 10:00:38 [Apache] Attempting to start Apache app. 10:00:38 [Apache] Status change detected: running </pre>

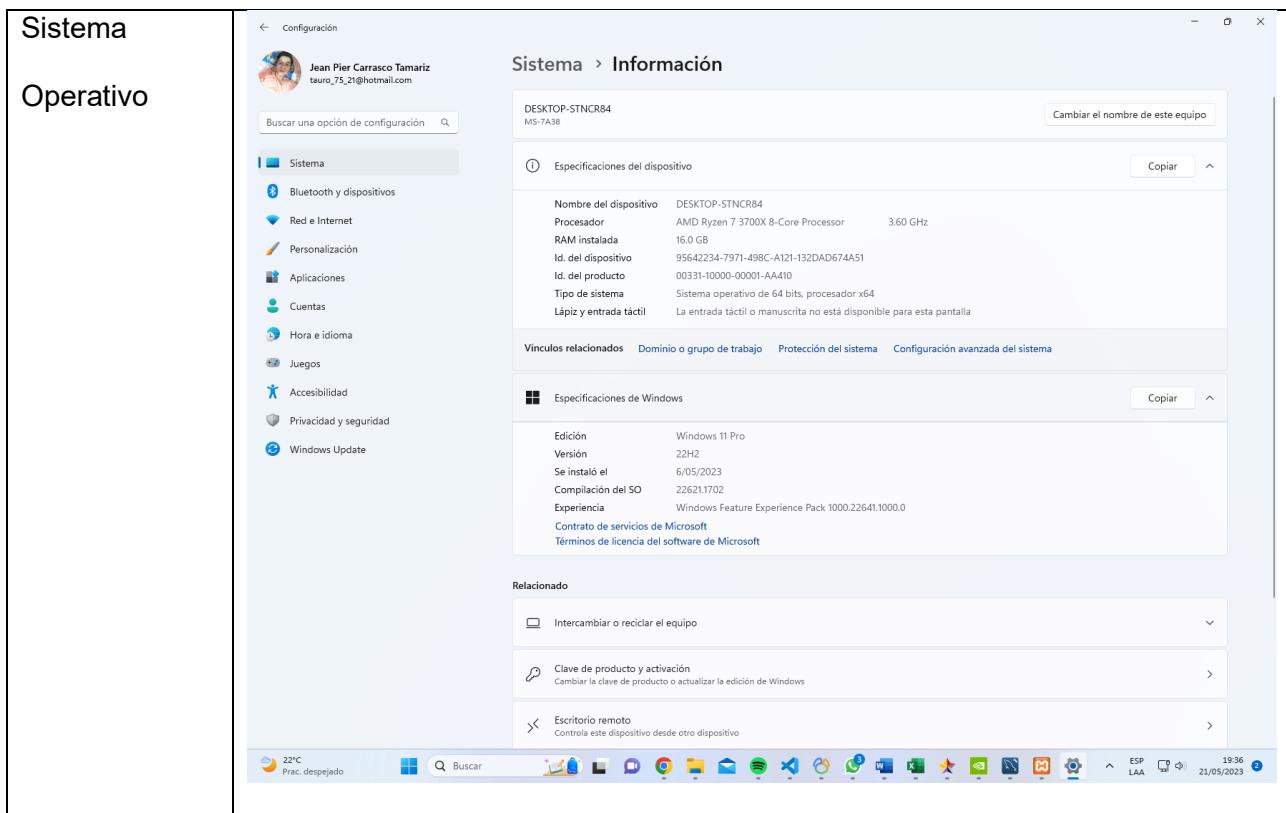
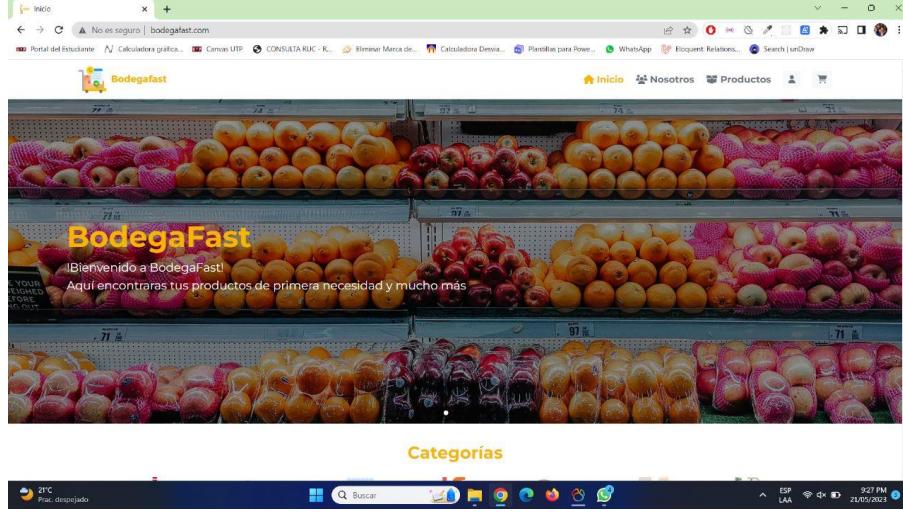
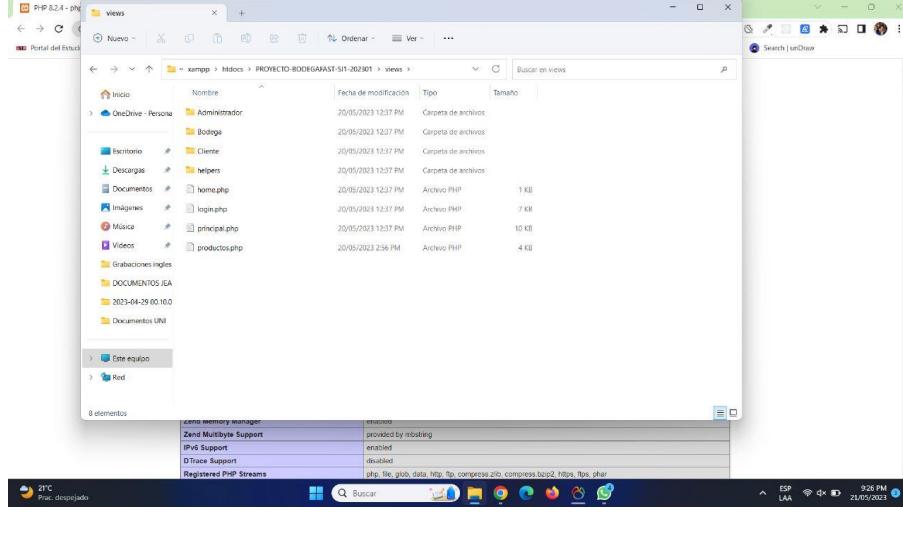


Tabla 4. Arquitectura de desarrollo Jean Pier

3.19. Arquitectura en producción

Navegador	
Front	

<h3>Back</h3>	
<h3>Base de Datos</h3>	<h3>Sistema Operativo</h3>

Tabla 5. Arquitectura de desarrollo en entorno de producción

3.20. Base de datos listar productos para la compra

Figura 60. Código de Base de datos listar productos de compra

```

1 CREATE DEFINER=`root`@`localhost` PROCEDURE `SP_R_T_PRODUCTOS_COMPRA` (IN _nombreProduct
VARCHAR(255), IN _ordenaras VARCHAR(255), IN _categorias VARCHAR(255), IN _marcas VARCHAR(
255))
2 BEGIN
3     /*AUTOR: JEAN PIER CARRASCO*/
4     /*SELECCIONAMOS LAS COLUMNAS NECESARIAS*/
5     SELECT p.id,p.nombre,p.img,p.precio_venta,p.stock,b.nombre AS bodega,m.nombre AS marca
FROM productos p
6     /*RELACIONAMOS CON LAS TABLAS CORRESPONDIENTES*/
7     INNER JOIN productos_categorias pc
8     ON pc.id_producto = p.id
9     INNER JOIN bodegas b
10    ON p.id_bodega = b.id
11    INNER JOIN marcas m
12    ON p.id_marca = m.id
13    /*APLICAMOS LAS CONDICIONALES*/
14    WHERE p.estado = 1 AND b.estado = 1
15    AND IF(_categorias != '',FIND_IN_SET(pc.id_categoria,_categorias) > 0,p.id LIKE CONCAT(
'%','_', '%'))
16    AND IF(_marcas != '',FIND_IN_SET(p.id_marca,_marcas) > 0,p.id LIKE CONCAT('%','_', '%'))
17    AND p.nombre LIKE CONCAT('%',_nombreProduct,'%')
18    /*ORDENAMOS DE ACUERDO AL FILTRO */
19    ORDER BY
20    IF(_ordenaras = 'precio-menor',p.precio_venta,NULL) ASC,
21    IF(_ordenaras = 'precio-mayor',p.precio_venta,NULL) DESC,
22    IF(_ordenaras = 'nombre-asc',p.nombre,NULL) ASC,
23    IF(_ordenaras = 'nombre-desc',p.nombre,NULL) DESC;
24 END

1
_idProductos VARCHAR(255))
2 BEGIN
3     /*AUTOR: JEAN PIER CARRASCO*/
4     /*SELECCIONAMOS LAS COLUMNAS NECESARIAS*/
5     SELECT p.id,p.nombre,p.img,p.precio_venta,p.stock,b.nombre AS bodega FROM productos p
6     /*RELACIONAMOS CON LAS TABLAS CORRESPONDIENTES*/
7     INNER JOIN bodegas b
8     ON p.id_bodega = b.id
9     /*APLICAMOS LAS CONDICIONALES*/
10    WHERE p.estado = 1 AND b.estado = 1
11    AND IF(_idProductos != '',FIND_IN_SET(p.id,_idProductos) > 0,p.id LIKE CONCAT('%','_', '%'))
12    /*ORDENAMOS POR NOMBRE*/
13    ORDER BY p.nombre;
14 END

```

3.21. Modelo para listar productos de compra

Figura 61. Código de Modelo para listar productos de compra

```
<?php
//AUTOR: JEAN PIER CARRASCO
//Establecemos el nombre de espacio donde se trabaja
namespace Models;
//Requerimos la conexion
require_once "Conexion.php";
//Usamos las clases de conexion
use Models\Conexion;
//Creamos la clase y lo extendemos a la Conexion
class Producto extends Conexion{
    //Definimos sus atributos de la clase
    private int $id;
    private int $idBodega;
    private int $idMarca;
    private string $idCategorias;
    private string $nombre;
    private string $descripcion;
    private float $stock;
    private float $stockMinimo;
    private float $precioCompra;
    private float $precioVenta;
    private float $descuento;
    private string $img;
    private int $estado;
```

Figura 62. Código de Modelo para listar productos de compra

```
//Definimos la funcion y le pasamos las variables necesarias
public function verProductosClientes(string $producto,string $categorias,string $marcas,
string $ordenar)
{
    //realizamos la conexion
    $cn = $this->conectar();
    //llamamos al procedimiento almacenado
    $stmt = $cn->prepare("CALL SP_R_T_PRODUCTOS_COMPRA($producto,$categorias,$marcas,$ordenar)");
    //colocamos las variables
    $stmt->bind_param("ssss",$producto,$ordenar,$categorias,$marcas);
    //ejecutamos la consulta
    $stmt->execute();
    //obtenemos los datos
    $rs = $stmt->get_result();
    //definimos un arreglo
    $result = [];
    //recorremos el resultado y lo llenamos en la variable
    while ($result[] = $rs->fetch_assoc());
    //eliminamos el ultimo registro que es nulo
    array_pop($result);
    //cerramos conección
    $stmt->close();
    //retornamos el resultado
    return $result;
}
```

3.22. Controlador para listar productos de compra

Figura 63. Código de **Controlador** para listar productos de compra

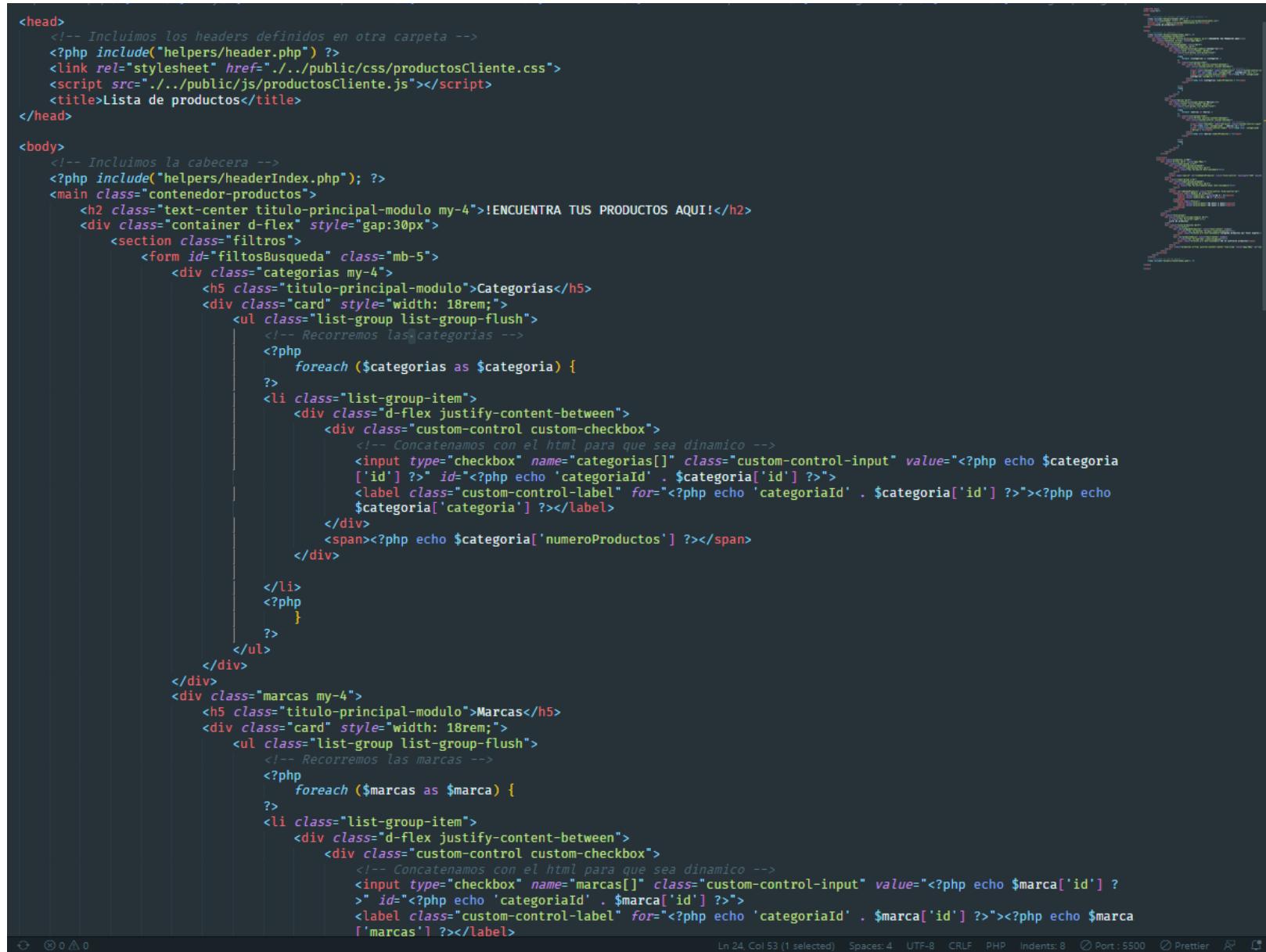
```

controllers / PaginaPrincipal.php > ...
1  <?php
2  //AUTOR: JEAN PIER CARRASCO
3  //le damos un nombre de espacio
4  namespace Controllers;
5  //requerimos los modelos
6  require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/UsuarioModel.php';
7  require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/CategoríaModel.php';
8  require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/MarcaModel.php';
9  //usamos los modelos
10 use Models\Usuario as ModelUsuario;
11 use Models\Categoría as CategoríaModel;
12 use Models\Marca as MarcaModel;
13 //creamos la clase
14 class PaginaPrincipal{
15     //cramos la funcion
16     public function indexHome()
17     {
18         //instanciamos el modelo usuario
19         $usuarioModel = new ModelUsuario();
20         //instanciamos el modelo categoria
21         $categoriaModel = new CategoríaModel();
22         //obtenemos los datos a traves de sus metodo
23         $data = $usuarioModel->obtenerDatosAutenticado();
24         $listaCategorías = $categoriaModel->mostrar();
25         //mostramos la vista principal
26         include("views/principal.php");
27     }
28     //cramos la funcion
29     public function indexVerProductos()
30     {
31         //instanciamos el modelo categoria
32         $categoriaModel = new CategoríaModel();
33         //instanciamos el modelo marca
34         $marcaModel = new MarcaModel();
35         //obtenemos los datos a traves de sus metodo
36         $categorias = $categoriaModel->obtenerCategoríasProductos();
37         $marcas = $marcaModel->obtenerMarcasProductos();
38         //mostramos la vista de los productos
39         include("views/productos.php");
40     }
41 }

```

3.23. Vista para listar productos de compra

Figura 64. Código de **Vista para listar productos de compra**



```

<head>
    <!-- Incluimos los headers definidos en otra carpeta -->
    <?php include("helpers/header.php") ?>
    <link rel="stylesheet" href="../../public/css/productosCliente.css">
    <script src="../../public/js/productosCliente.js"></script>
    <title>Lista de productos</title>
</head>

<body>
    <!-- Incluimos la cabecera -->
    <?php include("helpers/headerIndex.php"); ?>
    <main class="contenedor-productos">
        <h2 class="text-center titulo-principal-modulo my-4">!ENCUENTRA TUS PRODUCTOS AQUI!</h2>
        <div class="container d-flex" style="gap:30px">
            <section class="filtros">
                <form id="filtrosBusqueda" class="mb-5">
                    <div class="categorias my-4">
                        <h5 class="titulo-principal-modulo">Categorías</h5>
                        <div class="card" style="width: 18rem;">
                            <ul class="list-group list-group-flush">
                                <!-- Recorremos las categorias -->
                                <?php
                                    foreach ($categorias as $categoria) {
                                ?>
                                <li class="list-group-item">
                                    <div class="d-flex justify-content-between">
                                        <div class="custom-control custom-checkbox">
                                            <!-- Concatenamos con el html para que sea dinamico -->
                                            <input type="checkbox" name="categorias[]" class="custom-control-input" value="<?php echo $categoria['id'] ?>" id="<?php echo 'categoriaId' . $categoria['id'] ?>">
                                            <label class="custom-control-label" for="<?php echo 'categoriaId' . $categoria['id'] ?>"><?php echo $categoria['categoria'] ?></label>
                                        </div>
                                        <span><?php echo $categoria['numeroProductos'] ?></span>
                                    </div>
                                </li>
                                <?php
                            }
                        ?>
                    </ul>
                </div>
                <div class="marcas my-4">
                    <h5 class="titulo-principal-modulo">Marcas</h5>
                    <div class="card" style="width: 18rem;">
                        <ul class="list-group list-group-flush">
                            <!-- Recorremos las marcas -->
                            <?php
                                foreach ($marcas as $marca) {
                            ?>
                            <li class="list-group-item">
                                <div class="d-flex justify-content-between">
                                    <div class="custom-control custom-checkbox">
                                        <!-- Concatenamos con el html para que sea dinamico -->
                                        <input type="checkbox" name="marcas[]" class="custom-control-input" value="<?php echo $marca['id'] ?>" id="<?php echo 'marcaId' . $marca['id'] ?>">
                                        <label class="custom-control-label" for="<?php echo 'marcaId' . $marca['id'] ?>"><?php echo $marca['marcas'] ?></label>
                                    </div>
                                </div>
                            </li>
                        <?php
                    }
                </ul>
            </div>
        </div>
    </main>
</body>

```

Ln 24, Col 53 (1 selected) Spaces: 4 UTF-8 CRLF PHP Indents: 8 ⚡ Port: 5500 ⚡ Prettier ⚡

3.24. Revisiones de código

a. Caso de uso CRUD categorías

- Revisado por: Jean Pier Carrasco Tamariz
- Revisado a: Bryan Sánchez Luyo
- Correcciones:

En el archivo CategoriaModel.php se agregó los métodos get y set de los atributos nombre y estado, además en el archivo CategoríaController.php en la línea 2 y 5 se cambió el nombre de las clases por una con camelcase. En la línea 15 del mismo documento se agregó el retorno para la comunicación con el archivo http correspondiente, finalmente se separó el modal que estaba en la vista categoría para que posteriormente fuera llamado el modal en esa vista.

b. Caso de uso CRUD marcas

- Revisado por: Jean Pier Carrasco Tamariz
- Revisado a: Diego Marquéz Fernández
- Correcciones:

En el archivo MarcaModel.php se agregó los métodos get y set de los atributos nombre y estado, además se eliminaron métodos duplicados de agregar y actualizar en archivo MarcaController.php, también en el archivo Marca.php de la carpeta Http se agregó un caso más al switch para que funcionara el eliminar marca, finalmente en el procedimiento almacenado SP_D_T_MARCAS se modificó el eliminar por el actualizar.

3.25. Plan de pruebas

Tabla 6. Tabla de plan de pruebas

CUS	DA-NORMAL	RE-NORMAL	DA-ERROR	RE-ERROR
Agregar Producto Entregable 1	Nombre, descripción, categorías, marcas, costos, stock e imagen	Producto registrado correctamente	Precio de venta -1	El precio de venta debe ser mayor a cero
Agregar Bodega Entregable 1	Nombre, dirección, correo, teléfono, celular, localización y propietario	Bodega registrada correctamente y otorga una contraseña de acceso	Correo que contenga dos arrobas	El correo electrónico es invalido
Agregar Categoría Entregable 1	Nombre	Categoría registrada correctamente	Dejar en blanco el campo de la categoría	El nombre de la categoría no debe estar vacía
Agregar Marca Entregable 1	Nombre	Marca registrada correctamente	Digitar caracteres diferentes de letras y números	El nombre de la marca solo admite letras y/o números
Agregar un producto al carrito de compras Entregable 2	Cantidad	Producto agregado al carrito	Colocar -1	La cantidad del producto debe ser mayor a 0
Modificar la cantidad del producto en el carrito de compras Entregable 2	Cantidad	Cantidad modificada correctamente	Colocar 0	La cantidad del producto debe ser mayor a 0
Realizar una compra Entregable 2	Producto y cantidad	Compra realizada con éxito	Colocar 0 en cada detalle del producto	El total de la compra de ser mayor a 0
Colocar datos personales para la compra Entregable 2	Nombres, apellidos, dirección y celular	Compra realizada con éxito	Carácter vacío la dirección de envío	Por favor complete la dirección de envío

- Plan de ejecución

AGREGAR PRODUCTO – ENTREGABLE 1

Figura 65. DA - NORMAL

Agregar producto



Precio compra (S/) * Precio venta (S/) * Descuento (S/)

4.50	5.50	
------	------	--

Stock mínimo Stock *

0	10
---	----

[Atras](#) [Siguiente](#)

Figura 66. RE - NORMAL



Producto registrado correctamente

OK

Figura 67. DA - ERROR

Agregar producto



Precio compra (S/) * Precio venta (S/) * Descuento (S/)

5	-20	
---	-----	--

Stock mínimo Stock *

0	2
---	---

[Atras](#) [Siguiente](#)

Figura 68. RE - ERROR



El precio venta debe ser mayor a cero

OK

Figura 69. DA - NORMAL

Agregar Bodega X

RUC	Nombre *
<input type="text"/>	Boda Don Pedro
Correo *	<input type="text"/> @ bdonperdo@gmail.com
Direccion *	Av. Los alisos 120 - Los Olivos
Localización *	-12.0204747,-77.1606674,13
Telefono	Celular *
78451525	987845121
DNI Propietario *	Nombre Propietario *
25487845	Luis Mendoza Perez
<input checked="" type="checkbox"/> Estoy de acurdo con las condiciones	

Cerrar Agregar

Figura 71. DA - ERROR

Agregar Bodega X

RUC	Nombre *
<input type="text"/>	Boda Don Pedro
Correo *	<input type="text"/> @ bdonperdo@@gmail.com
Direccion *	Av. Los alisos 120 - Los Olivos
Localización *	-12.0204747,-77.1606674,13
Telefono	Celular *
78451525	987845121
DNI Propietario *	Nombre Propietario *
25487845	Luis Mendoza Perez
<input checked="" type="checkbox"/> Estoy de acurdo con las condiciones	

Cerrar Agregar

Figura 70. RE - NORMAL

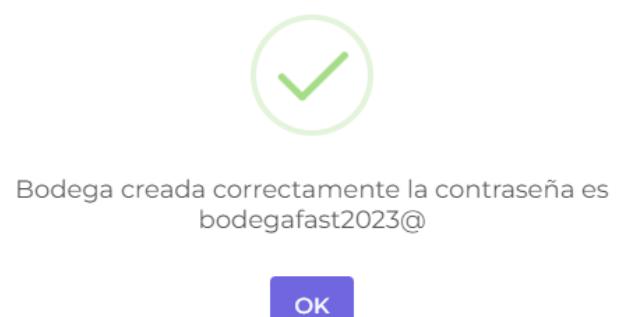


Figura 72. RE - ERROR

OK

AGREGAR CATEGORÍA – ENTREGABLE 1*Figura 73. DA - NORMAL*

Agregar Categoría X

Categoría

Cerrar Agregar

Figura 74. RE - NORMAL

Categoría agregada correctamente

Figura 75. DA - ERROR

Agregar Categoría X

Categoría

Cerrar Agregar

Figura 76. RE - ERROR

El nombre de la categoría no debe estar vacía

OK

AGREGAR MARCA – ENTREGABLE 1*Figura 77. DA - NORMAL*

Agregar Marca



Marca

Ajinomoto

Cerrar

Agregar

Figura 78. RE - NORMAL

Marca agregada correctamente

OK

Figura 79. DA - ERROR

Agregar Marca



Marca

MARCA@#%&

Cerrar

Agregar

Figura 80. RE - ERROR

El nombre de la marca solo acepta letras y/o
números

OK

AGREGAR PRODUCTO CARRITO – ENTREGABLE 2

Figura 81. DA - NORMAL



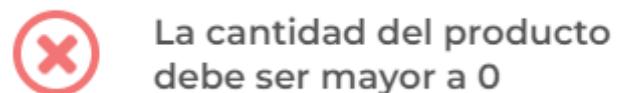
Figura 82. RE - NORMAL



Figura 83. DA - ERROR



Figura 84. RE - ERROR



MODIFICAR CANTIDADES CARRITO – ENTREGABLE 2

Figura 85. DA - NORMAL

Precio	Cantidad	Subtotal	Acciones
S/ 15.00	3	S/ 45.00	
S/ 2.20	1	S/ 2.20	
S/ 4.20	1	S/ 4.20	

Figura 86. RE - NORMAL



Cantidad modificada correctamente

Figura 87. DA - ERROR

Precio	Cantidad	Subtotal	Acciones
S/ 15.00	0	S/ 0.00	
S/ 2.20	1	S/ 2.20	
S/ 4.20	1	S/ 4.20	

Figura 88. RE - ERROR



La cantidad del producto debe ser mayor a 0

REGISTRAR COMPRA – ENTREGABLE 2

Figura 89. DA - NORMAL

Precio	Cantidad	Subtotal	Acciones
S/ 15.00	1	S/ 15.00	
S/ 2.20	1	S/ 2.20	
S/ 4.20	1	S/ 4.20	

Total: S/ 21.40

Figura 90. RE - NORMAL

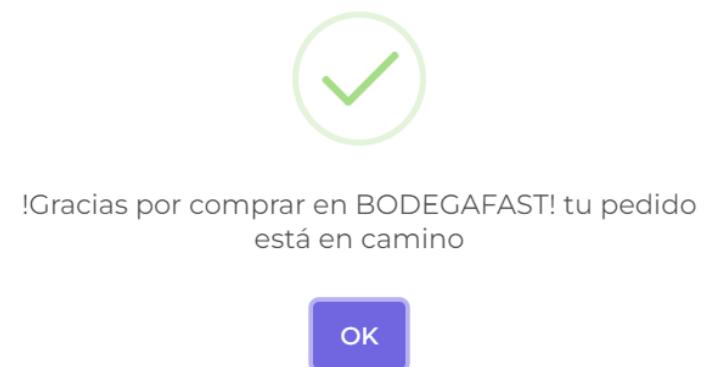
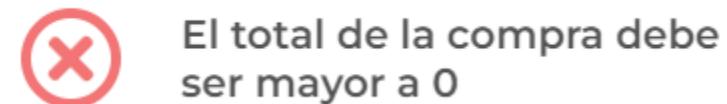


Figura 91. DA - ERROR

Precio	Cantidad	Subtotal	Acciones
S/ 15.00	0	S/ 0.00	
S/ 2.20	0	S/ 0.00	
S/ 4.20	0	S/ 0.00	

Total: S/ 0.00

Figura 92. RE - ERROR



DATOS PERSONALES – ENTREGABLE 2

Figura 93. DA - NORMAL

Datos personales

Nombres
jean pier

Apellidos
carra

Dirección de envío *

Av. Los alisos 120 - Los Olivos

Celular *

924777699

Figura 94. RE - NORMAL



!Gracias por comprar en BODEGAFAST! tu pedido está en camino



Figura 95. DA - ERROR

Datos personales

Nombres
jean pier

Apellidos
carra

Dirección de envío *



Celular *



Figura 96. RE - ERROR



Dirección

Por favor complete la dirección de envío



4. Entregable 3 – venta de productos para bodegas

4.1. Mockups (Facilidades de Uso) – 3 pasos – Tercer entregable

Figura 97. Venta de productos para bodega (Primer paso)

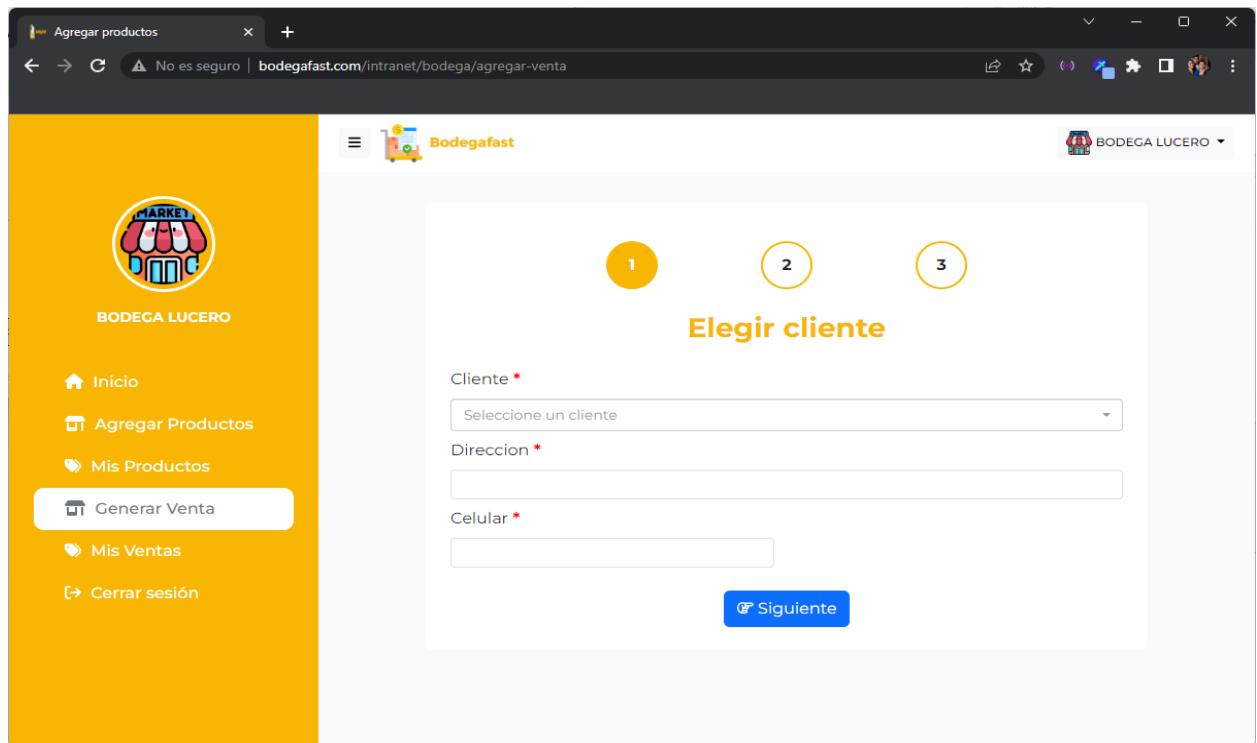


Figura 98. Venta de productos para bodega (Segundo paso)

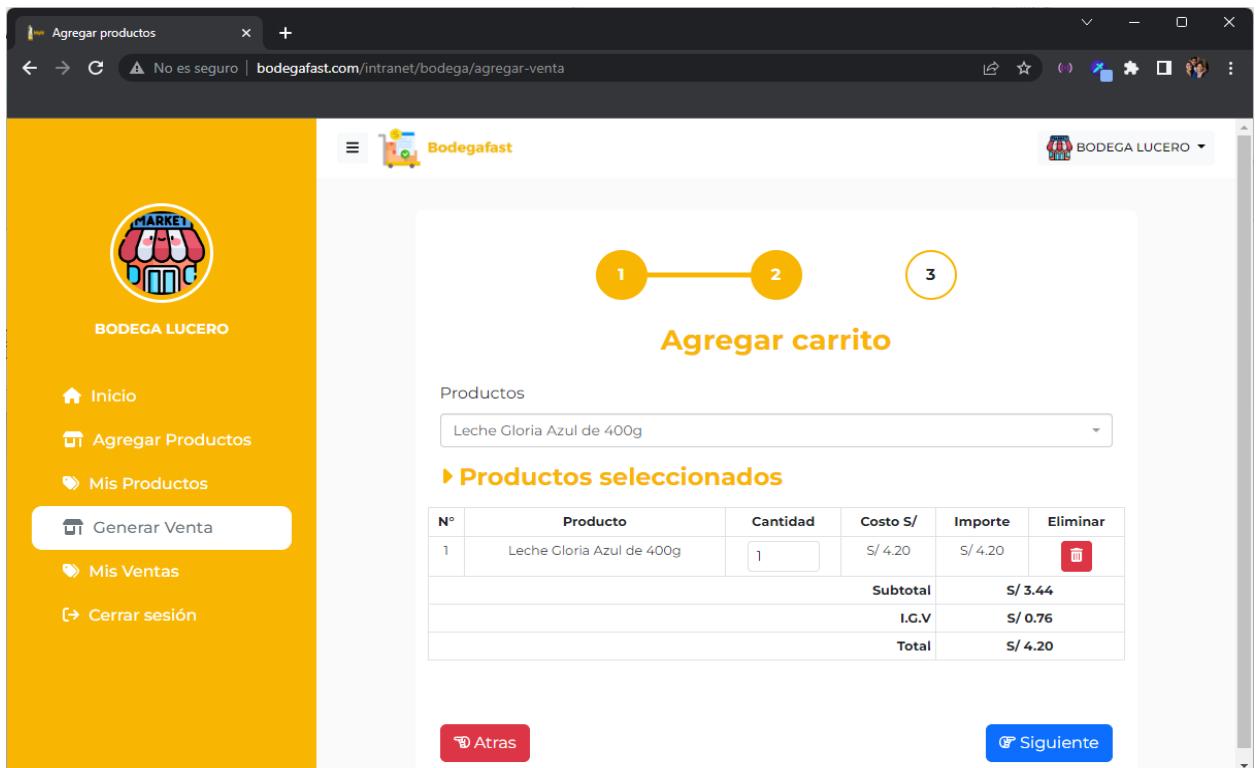


Figura 99. Venta de productos para bodega (Tercer paso)

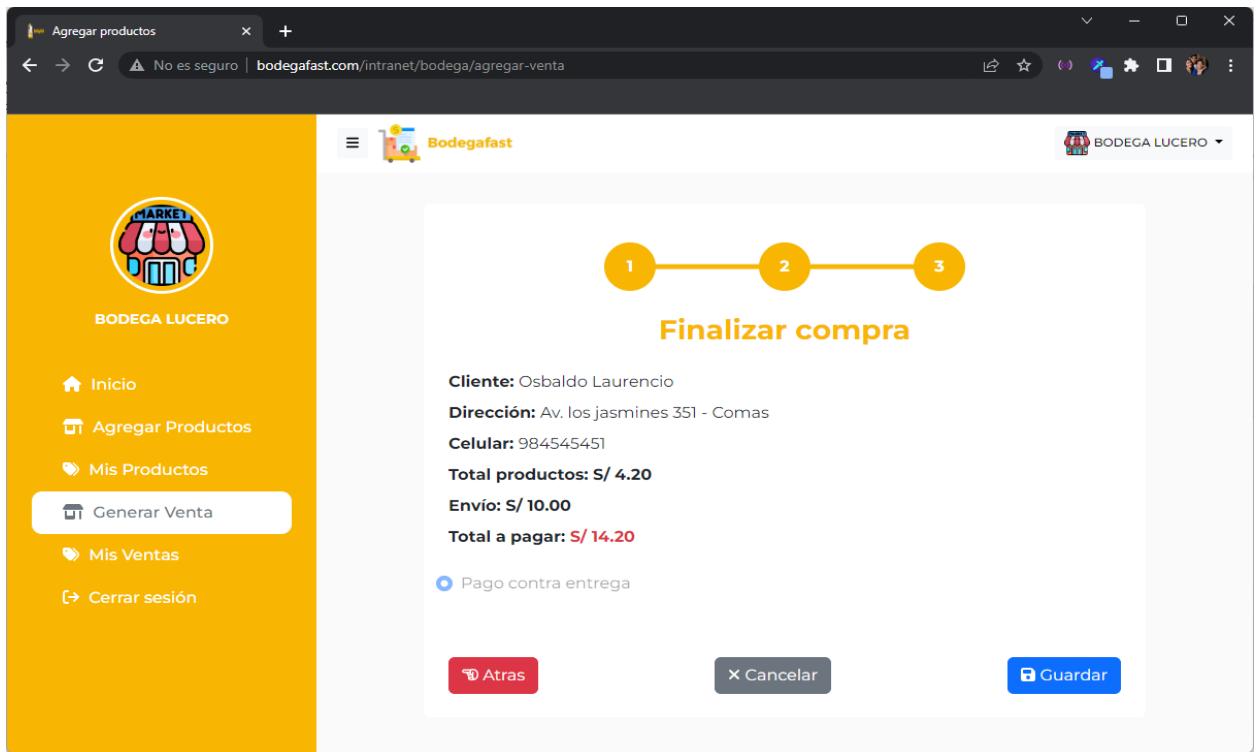
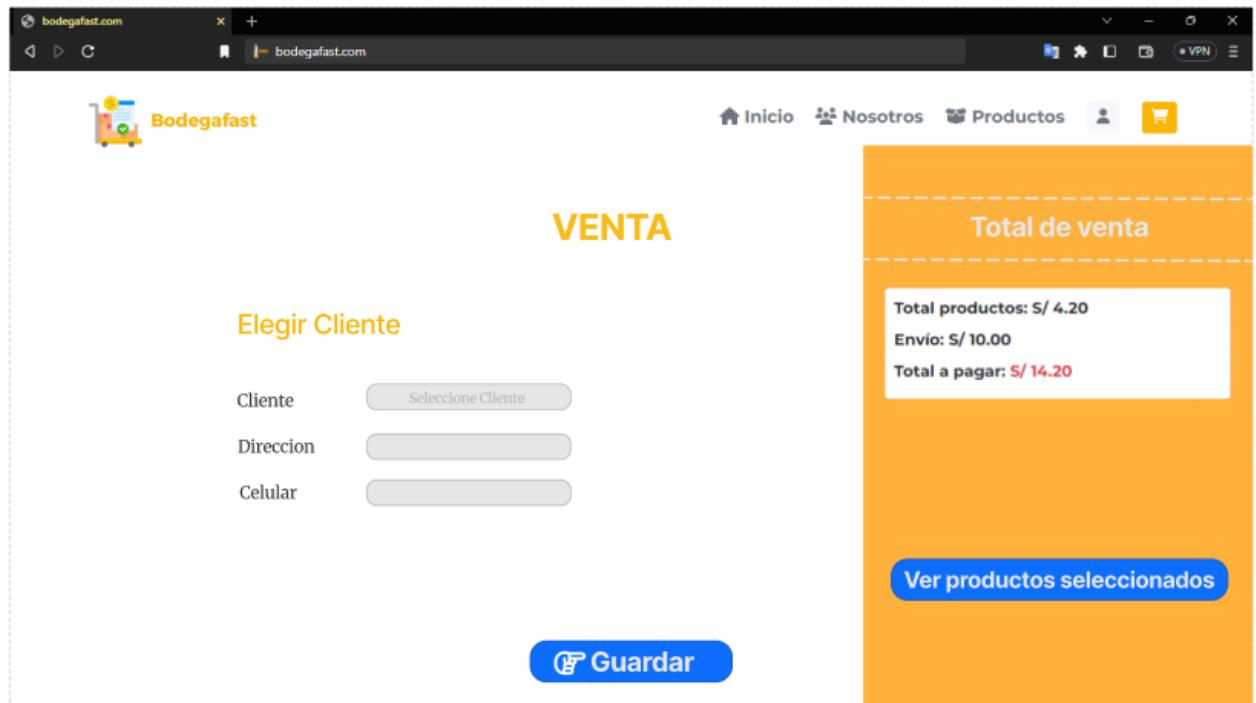


Figura 100. Venta de productos para bodega (Modelo 2)



4.2. GitHub actualizado

The screenshot shows a GitHub repository page for 'ploxomen/PROYECTO-BODEGAFAST-SI1-202301'. The repository has 4 branches and 0 tags. The most recent commit was made 41 minutes ago by 'JeanPier' with the message 'tercer entregable - venta por bodega - incluyendo comprobante por correo'. The commit hash is 6e44b34. The repository has 23 commits in total. The code section lists various files and their changes:

File	Description	Time Ago
Config	server	2 hours ago
Controllers	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
Http	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
Models	agregando tercer entregable	12 hours ago
db	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
public	agregando tercer entregable	12 hours ago
vendor	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
views	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
.htaccess	servidor	3 weeks ago
Router.php	modificando rutas	3 weeks ago
autoload.php	servidor	3 weeks ago
composer.json	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
composer.lock	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago
index.php	tercer entregable - venta por bodega - incluyendo comprobante por correo	41 minutes ago

The repository has no description, website, or topics provided. It has 0 stars, 1 watching, and 0 forks. There are no releases or packages published. The contributors are ploxomen and RaulMF11. The languages used are PHP (66.4%), JavaScript (26.9%), Hack (3.6%), and CSS (3.1%).

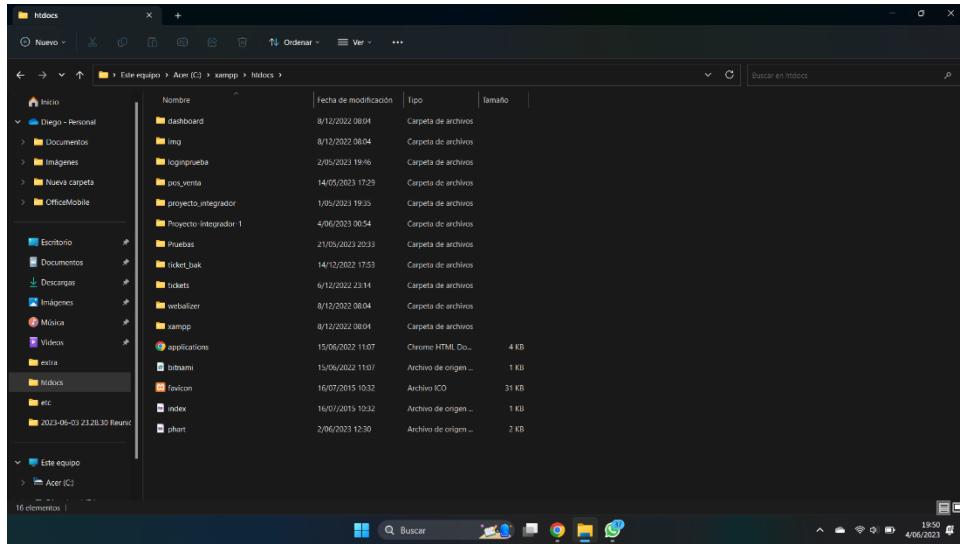
Link: <https://github.com/ploxomen/PROYECTO-BODEGAFAST-SI1-202301.git>

4.3. Creación de un ejecutable con extensión phar para comprimir nuestro proyecto

Considerar los siguientes pasos:

- Es necesario tener instalado Xampp con PHP versión 8 y MySQL server versión 8.
- Colocamos el archivo phar.php en la raíz principal de htdocs de Xampp

Figura 102. Colocacion del archivo phar.php



- Archivo phar.php que nos servirá para compilar el proyecto

Figura 103. Datos del archivo phar.php

```

phar
try
{
    //DEFINIMOS EL NOMBRE DEL ARCHIVO COMPRIMIDO
    $pharfile = "bodegafast.phar";
    // clean up
    if (file_exists($pharfile))
    {
        unlink($pharfile);
    }
    if (file_exists($pharfile . '.gz'))
    {
        unlink($pharfile . '.gz');
    }

    // CREAR EL PHAR
    $phar = new Phar($pharFile);
    // start buffering, Mandatory to modify stub to add shebang
    $phar->startBuffering();
    // Create the default stub from main.php entrypoint
    $stub = $phar->createDefaultStub('autoload.php');

    // Add the rest of the apps files
    $phar->buildFromDirectory(__DIR__ . '/proyecto-integrador-1');

    // Customize the stub to add the shebang
    //$stub = "#!/usr/bin/env php \n" . $stub;

    // Add the stub
    $phar->getStub($stub);
    $phar->stopBuffering();

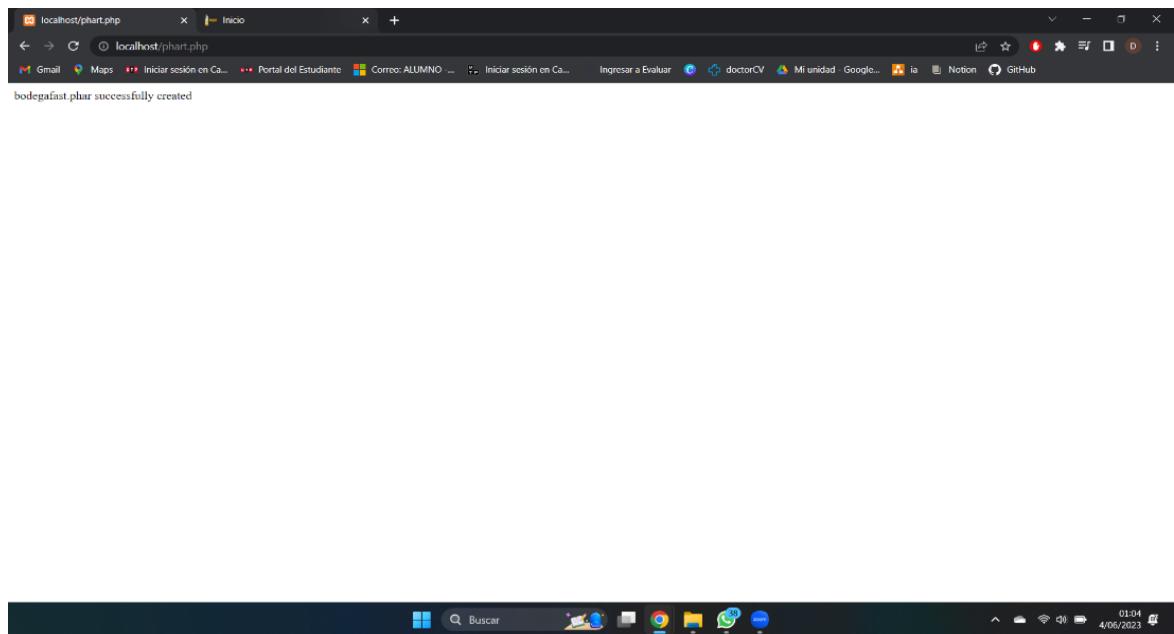
    // plus - compressing it into gzip
    $phar->compressFiles(Phar::GZ);

    # Make the file executable
}
Ln 1 Col 1

```

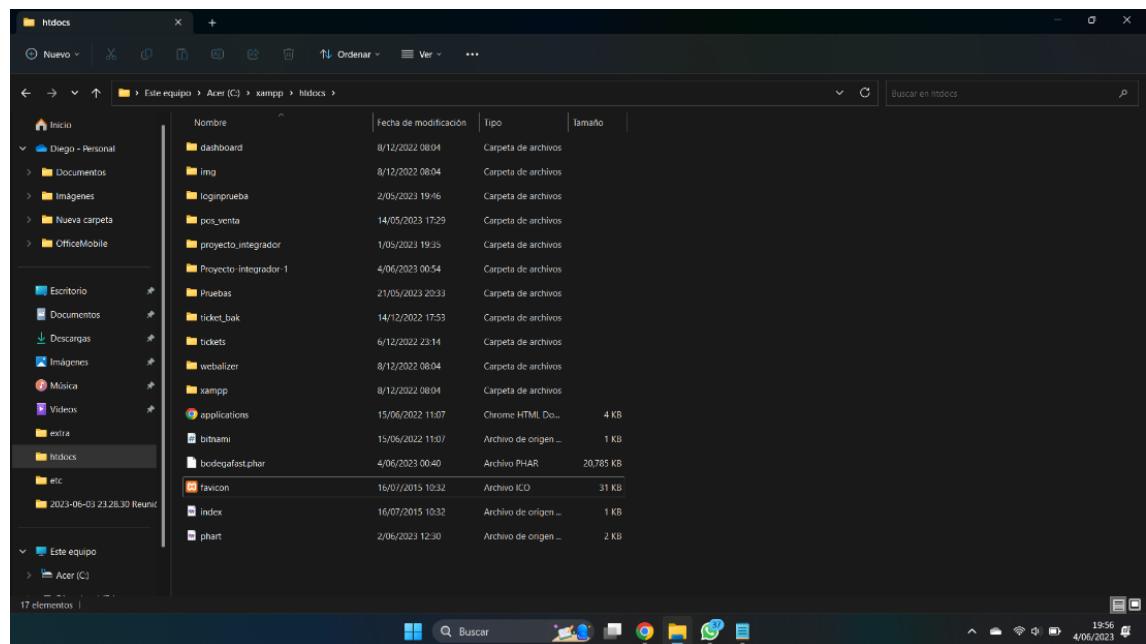
- Ejecutando el phar.php

Figura 104. Ejecucion del archivo phar.php



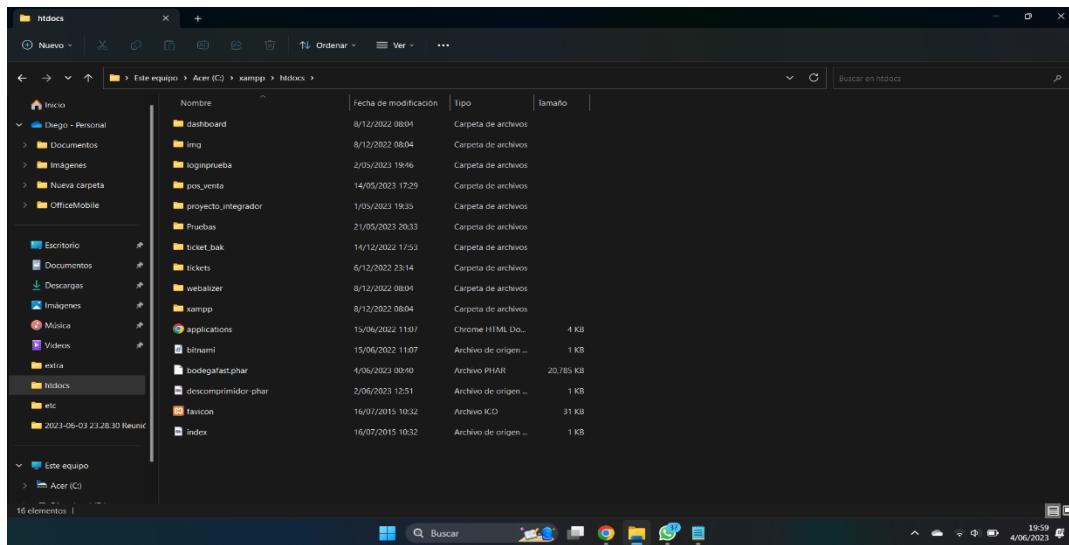
- Una vez ejecutado se crea el archivo compilado bodegafast.phar

Figura 105. Creacion del archivo ejecutable bodegafast.php



- Se copia el archivo en una nueva máquina y se instala el archivo descomprimidor.php en la raíz principal de htdocs de Xampp

Figura 106. Copia del archivo en una nueva maquina



- Archivo descomprimidor.php que nos servira para poder ejecutar el proyecto

Figura 107. Datos del archivo descomprimido.php

```

<?php
//die(sys_get_temp_dir());
$pharfile = "bodegafast.phar";
$extractPath = "Proyecto_integrador-1/";

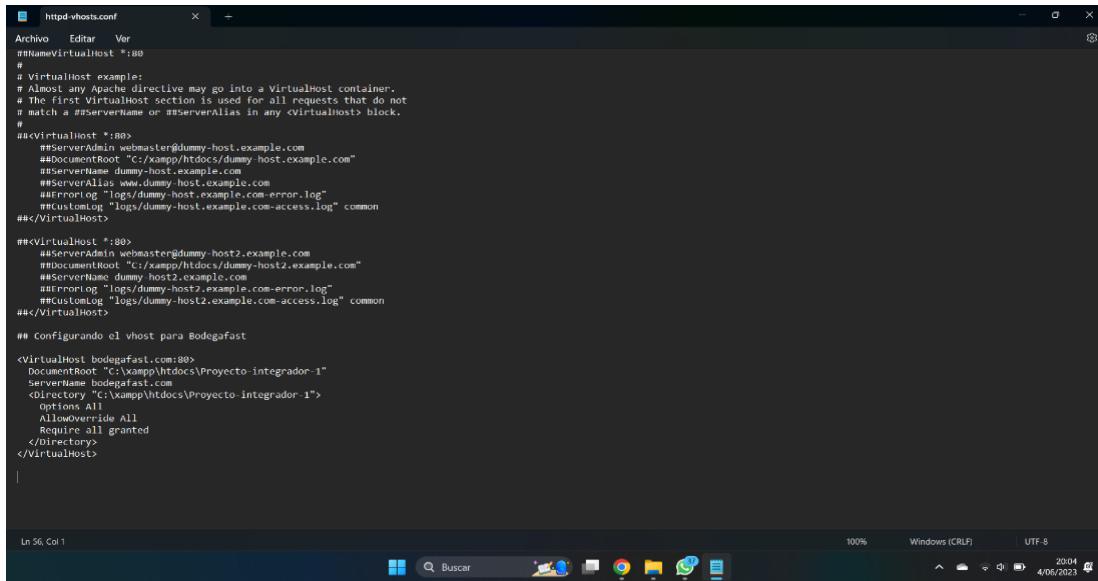
$phar = new Phar($pharfile);
$phar->extractTo($extractPath);

header("location: http://bodegafast.com");
?>

```

- Antes de ejecutar el descomprimidor se deberá hacer algunas configuraciones en la nueva máquina, primero se configura el vhost de apache en Xampp

Figura 108. Datos del archivo phar.php



```

httpd-vhosts.conf
Archivo Editar Ver
##NameVirtualHost *:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a #serverName or #serverAlias in any <VirtualHost> block.
#
##<VirtualHost *:80>
#ServerAdmin webmaster@dummy-host.example.com
#DocumentRoot "C:/xampp/htdocs/dummy-host.example.com"
#ServerName dummy-host.example.com
#ErrorLog "logs/dummy-host.example.com-error.log"
#CustomLog "logs/dummy-host.example.com-access.log" common
##</VirtualHost>

##<VirtualHost *:80>
#ServerAdmin webmaster@dummy-host2.example.com
#DocumentRoot "C:/xampp/htdocs/dummy-host2.example.com"
#ServerName dummy-host2.example.com
#ErrorLog "logs/dummy-host2.example.com-error.log"
#CustomLog "logs/dummy-host2.example.com-access.log" common
##</VirtualHost>

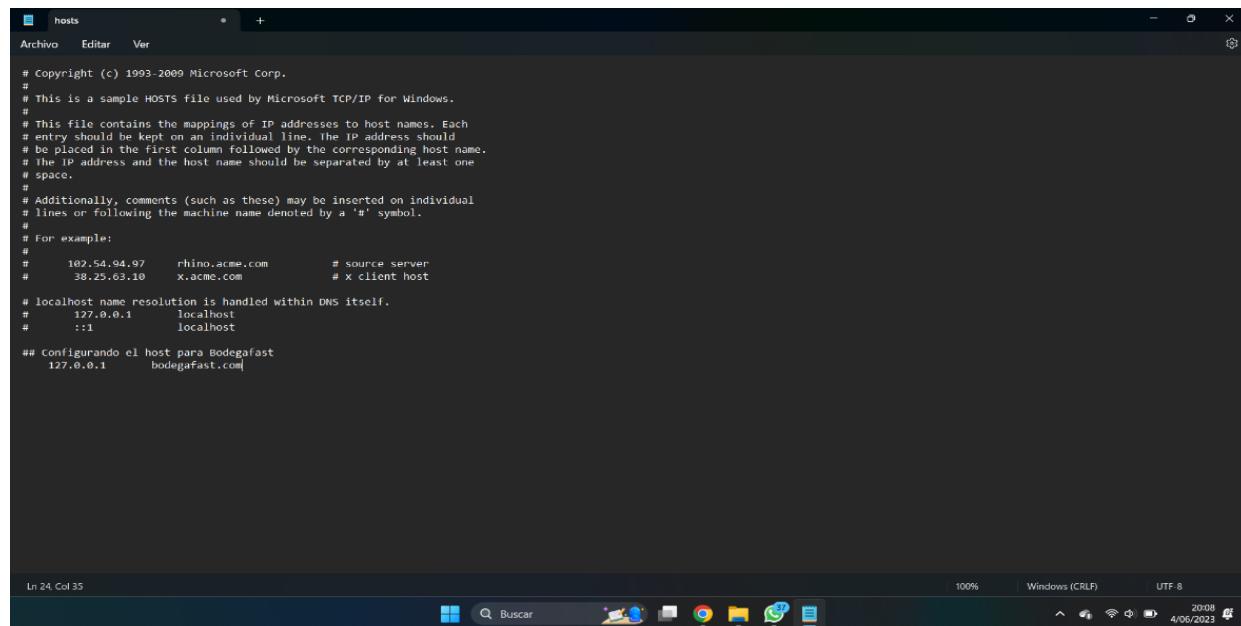
## Configurando el vhost para Bodegafast
<VirtualHost bodegafast.com:80>
    DocumentRoot "C:/xampp/htdocs/Proyecto-integrador-1"
    ServerName bodegafast.com
    <Directory "C:/xampp/htdocs/Proyecto-integrador-1">
        Options All
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
|

```

- Luego se configura el host dentro de la siguiente ruta:

C:\Windows\System32\drivers\etc

Figura 109. Configuración del Host



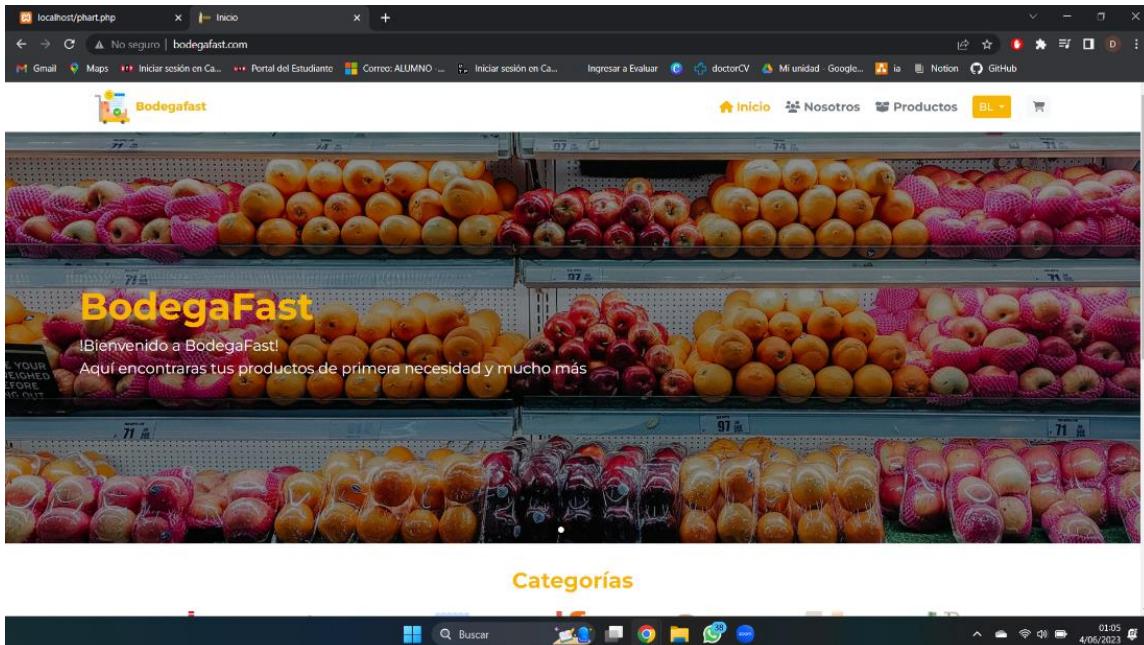
```

hosts
Archivo Editar Ver
#
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97    rhino.acme.com        # source server
#      38.25.63.10    x.acme.com            # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1        localhost
#
## Configurando el host para Bodegafast
127.0.0.1        bodegafast.com

```

-Desactivamos y activamos nuevamente Xampp y luego ejecutamos el archivo descomprimidor.php para que se ejecute el proyecto

Figura 110. Configuracion del Host



4.4. Notificación por correo electrónico cada vez que se realiza una compra por parte del cliente o de la bodega.

Considerar los siguientes pasos:

1. Es necesario tener instalado composer para poder descargar librerías en nuestro proyecto.
2. Instalamos la librería de PHPMAILER con el comando “composer require phpmailer/phpmailer” dentro de nuestro directorio donde está el proyecto.
3. Creamos nuestro controlador Correo.php dentro de la carpeta Controllers e incluimos la librería para poder hacer usos de sus métodos y clases.

Figura 111. Creacion de un nuevo controlador

```

Correo.php ×
Controllers > Correo.php > ...
1  <?php
2  //Autor : Jean Pier Carrasco Tamariz
3  namespace Controllers;
4  //Llamamos a la libreria PHPMailer
5  require_once $_SERVER['DOCUMENT_ROOT'] . '/vendor/autoload.php';
6  use PHPMailer\PHPMailer\PHPMailer;
7  use PHPMailer\PHPMailer\Exception;
8  //Creamos la clase

```

4. Creamos nuestra clase Correo y dentro de ella una función de configuración para que la librería tenga acceso y enviar los correos, además creamos otra función el cual se encargara de enviar el correo a nuestros clientes quienes hayan comprado.

Figura 112. Creacion de la clase correo

```

class Correo{
    //Definimos la funcion de configuracion
    public function hostCorreoConfiguracion()
    {
        //Llamamos a phpMailer
        $mail = new PHPMailer(true);
        try {
            //Decimos que el protocolo es SMTP
            $mail->isSMTP();
            //Deshabilitamos el debug
            $mail->SMTPDebug = 0;
            // Colocamos el host de gmail
            $mail->Host = 'smtp.gmail.com';
            //Habilitamos la autenticacion con gmail
            $mail->SMTPAuth = true;
            // Colocamos el correo del cual sera enviado
            $mail->Username = 'jeanpi.jpct@gmail.com';
            // colocamos la contraseña el cual es una encriptada
            $mail->Password = 'yhtmkobrranuecc';
            // Encriptamos la contraseña
            $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
            //Definimos el puerto
            $mail->Port = 465;

            //Enviamos desde el correo asignado
            $mail->setFrom('jeanpi.jpct@gmail.com', 'BODEGAFAST');
            //Retornamos el mail
            return $mail;
        } catch (Exception $e) {
            return ['error' => $mail->ErrorInfo];
        }
    }

    public function enviarCorreoCompra(string $asunto, string $correoEnvio, string $nombreCorreto, string $body, string $bodyShort)
    {
        //Obtenemos el mail
        $mail = $this->hostCorreoConfiguracion();
        //Añadimos nuestro logo
        $mail->addAttachment($_SERVER['DOCUMENT_ROOT'].'/Public/img/logo.png', 'bodegafast.jpg');
        //añadimos el correo a enviar
        $mail->addAddress($correoEnvio, $nombreCorreto);
        //Habilitamos el html
        $mail->isHTML(true);
        //Añadimos el asunto
        $mail->Subject = $asunto;
        //Definimos UTF8 Para no tener problemas con caracteres especiales
        $mail->CharSet = "UTF-8";
        //Añadimos el cuerpo
        $mail->Body = $body;
        //Añadimos el pequeño cuerpo
        $mail->AltBody = $bodyShort;
        //Enviamos el correo
        $mail->send();
        return ['success' => 'Correo enviado correctamente'];
    }
}

```

5. En nuestro controlador donde tenemos la función de agregar venta llamamos al archivo Correo.php.

Figura 113. Funcion de agregar ventas

```
<?php
namespace Controllers\Cliente;
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/ProductoModel.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/UsuarioModel.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/VentasModel.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/Controllers/Correo.php';

use Models\Usuario as ModelUsuario;
use Models\Producto as ModelProducto;
use Models\Ventas as VentasModel;
use Controllers\Correo;
```

6. En nuestra función donde generamos la venta llamamos al método del correo siempre y cuando se registre correctamente a nuestra base de datos.

Figura 114. Generacion de venta llamando al metodo del correo

```
public function agregarCompraCliente(array $datos)
{
    //OBTEMOS EL CARRITO DE COMPRAS
    $detalleVenta = isset($_COOKIE['carrito_compras']) ? json_decode($_COOKIE['carrito_compras'],true) : [];
    $subtotal = 0;
    //OBTEMOS EL MODELO DEL PRODUCTO
    $modeloProducto = new ModelProducto;
    //FILTRAMOS LOS PRODUCTOS
    $idProductos = implode(", ",array_column($detalleVenta,"idProducto"));
    //OBTEMOS LOS DATOS DE LA DB PARA COMPRAR
    $productos = $modeloProducto->verProductosClientesCarrito($idProductos);
    //RECORREMOS PARA FILTRAR
    foreach ($productos as $kp => $vp) {
        //FILTRAMOS EL ARREGLO
        $filtroCarrito = array_filter($detalleVenta,function($valCarrito)use($vp){
            return $valCarrito['idProducto'] == $vp['id'];
        });
        if(empty($filtroCarrito)){
            continue;
        }
        //DEFINIMOS LA CANTIDAD Y EL SUBTOTAL
        $productos[$kp]['cantidad'] = $detalleVenta[key($filtroCarrito)][['cantidad']];
        $productos[$kp]['sub_total'] = $detalleVenta[key($filtroCarrito)][['cantidad']] * $vp['precio_venta'];
        $subtotal += $productos[$kp]['sub_total'];
    }
}
```

```

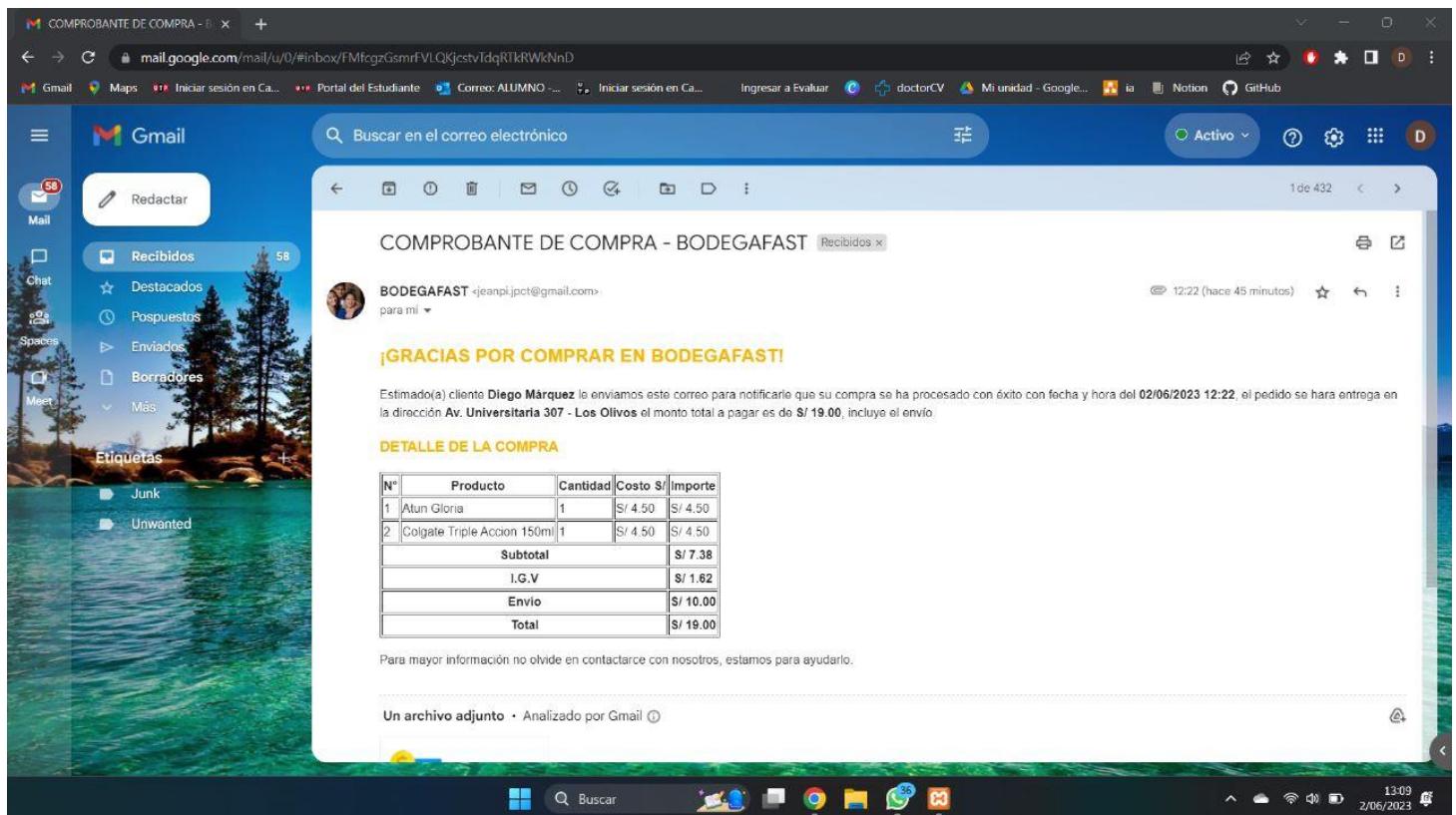
//DEFINIMOS LAS VARIABLES PARA NUESTRA COMPRA
$igv = floatval(0.18 * $subtotal);
$envio = floatval($datos['envio']);
$total = floatval($subtotal + $envio);
$ventaModel = new VentasModel();
$ventaModel->setIdCliente($data['idAccesoRol']);
$ventaModel->setDireccion($datos['direccion']);
$ventaModel->setCelular($datos['celular']);
$ventaModel->setMetodoEnvio("DELIVERY");
$ventaModel->setMetodoPago("EFECTIVO");
$ventaModel->setResponsableVenta("CLIENTE");
$ventaModel->setEnvio($datos['envio']);
$ventaModel->setSubtotal($subtotal);
$ventaModel->setTotal($total);
$ventaModel->setIgv($igv);
$ventaModel->setDetalleVentas(json_encode($productos));
$resultado = $ventaModel->agregarVenta();

//SI TODO ESTA OK SE ENVIA EL CORREO
if($resultado){
    //SE DEFINE LA ZONA HORARIA
    date_default_timezone_set('America/Bogota');
    $direccion = $datos['direccion'];
    $nombreCompleto = $data['nombres'] . ' ' . $data['apellidos'];
    //OBTENEMOS EL PHP EN TEXTO DEL CORREO
    ob_start();
    include_once $_SERVER['DOCUMENT_ROOT'] . '/Views/Cliente/correoCompra.php';
    //LO ALMACENAMOS EN UNA VARIABLE
    $contenido_html = ob_get_clean();
    //INSTANCIAMOS EL CORREO
    $correo = new Correo;
    //ENVIAMOS EL CORREO AL DESTINATARIO CON LOS CAMPOS NECESARIOS
    $correo->enviarCorreocompra('COMPROBANTE DE COMPRA - BODEGAFAST',$data['correo'],
    $nombreCompleto,$contenido_html,';Gracias por comprar en BODEGAFAST!');
}
//RETORNAMOS EL RESULTADO
return !$resultado ? ['error' => 'Error al agregar una venta'] : ['success' => 'Venta generada con éxito'];

```

7. Listo tendremos el correo enviado al usuario o cliente correspondiente, los correos se envían en los dos entregables, al realizar la compra por parte del cliente y al registrar una venta para un cliente por parte de la bodega.

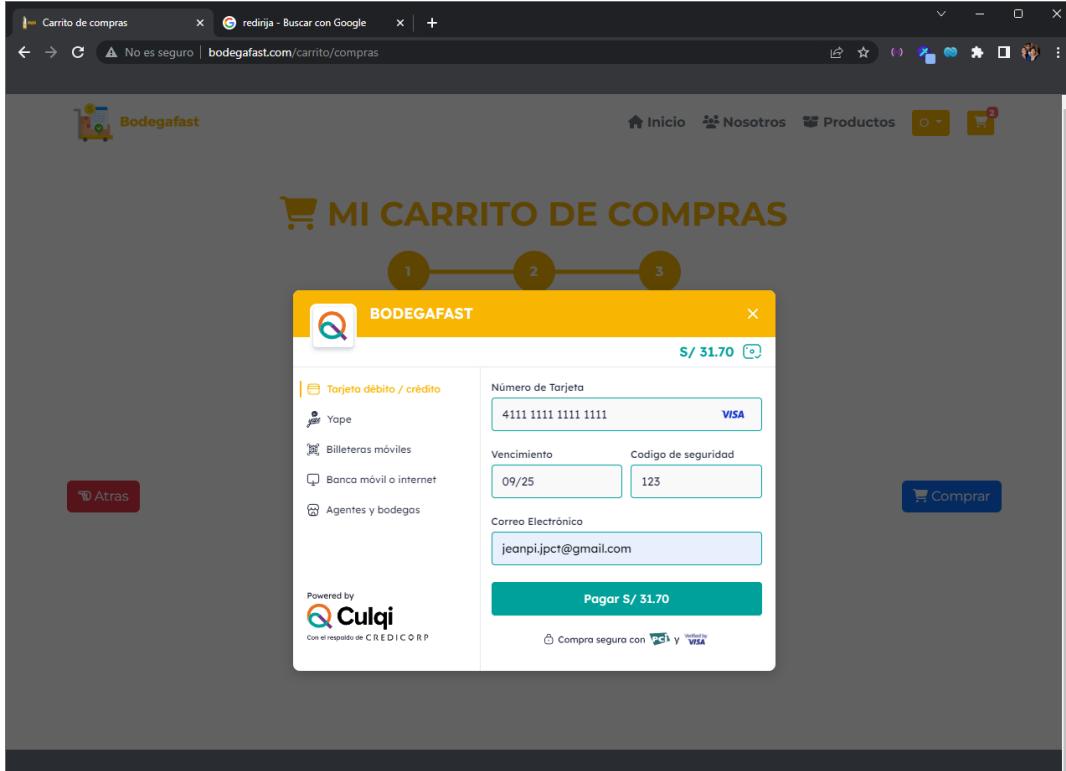
Figura 115. Comprobante de compra de BodegaFast



5. Entregable 4 – Métodos de pago y envío

5.1. Mockup método de pago

Figura 116. Método de Pago



5.2. Uso de Culqi como medio de pago – frontend

Figura 117. Medio de Pago - fronted

```
//Llave pública que te proporciona culqi
Culqi.publicKey = 'pk_test_5363217173e180bb';
//Opciones para habilitar la variedad de metodos de pago
Culqi.options({
    lang: "auto",
    installments: false,
    paymentMethods: {
        tarjeta: true,
        bancaMovil: true,
        agente: true,
        billetera: true,
        cuotealo: true,
        yape: true
    },
    //Estilos para darle color a la pasarela de pagos
    style: {
        bannerColor: '#F8B602',
        menuColor: '#F8B602',
        linksColor: '#F8B602'
    }
});
```

5.3. Creación de las órdenes en el controlador para los métodos de pago

Figura 118. Controlador para los métodos de pago

```

public function verificarAutenticacionCompra()
{
    error_reporting(0);
    //Obtenemos el carrito de compras
    $tokenBodegafast = isset($_COOKIE['token_bodegafast']) ? json_decode($_COOKIE['carrito_compras'],true) : [];
    $response = ['token' => false, 'rol' => null, 'nombres' => null, 'apellidos' => null];
    if(!empty($tokenBodegafast)){
        $total = 0;
        $response['token'] = true;
        $usuarioModel = new ModelUsuario();
        //verificamos el usuario autenticado
        $data = $usuarioModel->obtenerDatosAutenticado();
        $response['rol'] = $data['rol'];
        $response['nombres'] = $data['nombres'];
        $response['apellidos'] = $data['apellidos'];
        $idProductos = "";
        //asignamos los costos a los productos
        foreach ($tokenBodegafast as $key => $value) {
            $idProductos .= $value['idProducto'];
            $idProductos .= ($key + 1) != count($tokenBodegafast) ? "," : "";
        }
        //calculamos el subtotal con el total
        $modeloProducto = new ModelProducto();
        $productos = $modeloProducto->verProductosClientesCarrito($idProductos);
        foreach ($productos as $kp => $vp) {
            $filtroCarrito = array_filter($tokenBodegafast,function($valCarrito)use($vp){
                return $valCarrito['idProducto'] == $vp['id'];
            });
            if(empty($filtroCarrito)){
                continue;
            }
            $productos[$kp]['cantidad'] = $tokenBodegafast[key($filtroCarrito)]['cantidad'];
            $productos[$kp]['subtotal'] = $tokenBodegafast[key($filtroCarrito)]['cantidad'] * $vp['precio_venta'];
            $total += $productos[$kp]['subtotal'];
        }
        $response['total'] = $total;
        //generamos una orden para los métodos de pago
        $culqi = new \Culqi\Culqi(array('api_key' => $this->SECRET_KEY));
        $order = $culqi->Orders->create(
            array(
                "amount" => ($total + 10) * 100,
                "currency_code" => "PEN",
                "description" => 'Venta BODEGAFAST',
                "order_number" => uniqid(),
                "client_details" => array(
                    "first_name" => $data['nombres'],
                    "last_name" => $data['apellidos'],
                    "email" => $data['correo'],
                    "phone_number" => $data['celular']
                ),
                "expiration_date" => time() + 24*60*60// Orden con un dia de validez
            )
        );
        $response['orden'] = $order;
    }
    re //Recepcion de las órdenes para la asignacion del monto a pagar
    Culqi.settings({
        title: 'BODEGAFAST',
        currency: 'PEN',
        amount: response.orden.amount,
        order: response.orden.id
    });
    //Validacion de métodos de pago
    Culqi.validationPaymentMethods();
    paymentTypeAvailable = Culqi.paymentOptionsAvailable;
    return
}

```

5.4. Integración de culqi en nuestro método “agregarCompra” en el controlador Cliente

Figura 119. Integración de culqi

```
//Creamos nuestra venta en culqi
$culqi = new \Culqi\Culqi(array('api_key' => $this->SECRET_KEY));
$charge = $culqi->Charges->create(
    array(
        "amount" => ($total * 100),
        "capture" => true,
        "currency_code" => "PEN",
        "description" => "Compra BODEGAFAST",
        "email" => $datos['email'],
        "installments" => 0,
        "source_id" => $datos['token']
    )
);
```

5.5. Mockup – Mis ventas para el rol bodegas con reportes

Figura 120. Mis ventas para el rol de bodegas

Nº Venta	Fecha	Cliente	Celular	Dirección	Nº Productos	Cantidad	Importe	I.G.V
00023	02/06/2023	Yosimar Ramirez	987845123	AV. Los libertadores 14 - Carabayllo	1	3	S/ 19.68	S/ 4.32
00024	14/06/2023	Yosimar Ramirez	987845123	AV. Los libertadores 14 - Carabayllo	2	2	S/ 5.25	S/ 1.15
00025	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	2	5	S/ 24.19	S/ 5.31
00026	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	14		S/ 0.00	S/ 0.00
00027	14/06/2023	Osbaldo	984545451	Av. los jasmines 351 - Comas	1	1	S/ 3.69	S/ 0.81

5.6. Mockup – Reporte detalle de ventas

Figura 121. Reporte detallado de ventas

Bodegafast

Detalle de ventas desde 16/05/2023 hasta 15/06/2023

Nº Venta	Fecha Venta	Cliente	Método Pago	Producto	Precio	Cantidad	Subtotal
00025	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	3	S/13.50
00027	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	2	S/16.00
00028	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00029	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00030	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00031	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00032	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00033	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	2	S/16.00
00034	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	1	S/8.00
00035	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	1	S/8.00
00036	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00037	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	S/4.50	1	S/4.50
00038	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	1	S/8.00
00039	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	2	S/16.00
00040	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	S/8.00	2	S/16.00
00041	14/06/2023	Osbaldo Laurencio	CARD	Colgate Luminus White 150ml	S/6.70	3	S/20.10
00042	14/06/2023	Osbaldo Laurencio	CARD	Colgate Triple Accion 150ml	S/4.50	1	S/4.50
00043	14/06/2023	Osbaldo Laurencio	CARD	Colgate Triple Accion 150ml	S/4.50	2	S/9.00

5.7. Integración del reporte en el controlador Venta

Figura 122. Integración del reporte

```
public function reporteVenta(){
    //verificar la autenticacion del usuario
    $usuarioModel = new UsuarioModel();
    $data = $usuarioModel->obtenerDatosAutenticado();
    if (empty($data)) {
        //si no esta autenticado que le redirija al login
        header("location: /login");
        die();
    }
    if (!in_array($data['rol'], [$usuarioModel->rolBodega])) {
        //Si no esta con el rol bodega que le mande al inicio
        header("location: /intranet/inicio");
        die();
    }
    //se llama a las ventas
    $ventaModel = new VentasModel();
    //se llama a la libreria del reporte
    $dompdf = new Dompdf();
    $fechaInicio = $_POST['fechaInicio'];
    $fechaFin = $_POST['fechaFin'];
    //se obtiene las ventas
    $ventas = $ventaModel->verVentasPorBodega($data['idAccesoRol'], $fechaInicio, $fechaFin);
    foreach ($ventas as $k=>$venta) {
        $ventaModel->setId($venta['id']);
        //se obtiene el detalle de las ventas
        $ventas[$k]['productos'] = $ventaModel->verDetalleVentas();
    }
    //se incluye una vista html
    ob_start();
    include_once $_SERVER['DOCUMENT_ROOT'] . '/Views/Bodega/reportes/detalleVenta.php';
    $html = ob_get_clean();
    //Obtencion de la vista
    $dompdf->loadHtml($html);
    //definimos el papel horizontal
    $dompdf->setPaper('A4', 'landscape');
    //renderizamos en el navegador
    $dompdf->render();
    $dompdf->stream("reporte_ventas.pdf", array("Attachment" => false));
}
```

5.8. Mockup actualización de stock y precio de venta

Figura 123. Modulo historial Producto

The screenshot shows a web-based application interface for managing product history. On the left, there's a sidebar with a logo for 'BODEGA LUCERO' and a list of navigation items: Inicio, Agregar Productos, Mis Productos, Generar Venta, Mis Ventas, and Historial Producto (which is currently selected). The main content area has a header 'Bodegafast' and a sub-header 'Mis productos'. A modal window titled 'Editar Stock - Precio venta' is open, showing fields for 'Producto' (Leche Gloria Azul de 400g), 'Precio venta' (4.50), 'Descuento' (1.00), and 'Stock' (10.00). Below the modal is a section titled 'Historial de modificaciones' with a table showing three rows of modification history:

Nº	Fecha y Hora	Cantidad	Precio Venta	Descuento
1	14/06/2023 04:02 PM	3	S/ 4.50	S/1.00
2	14/06/2023 04:01 PM	15	S/ 4.50	S/1.00
3	14/06/2023 03:42 PM	222	S/12.00	S/ 0.00

At the bottom of the table, it says 'Mostrando 1 a 3 de 3 registros' and has navigation buttons 'Anterior' and 'Siguiente'.

6. Alternativas de Dashboard

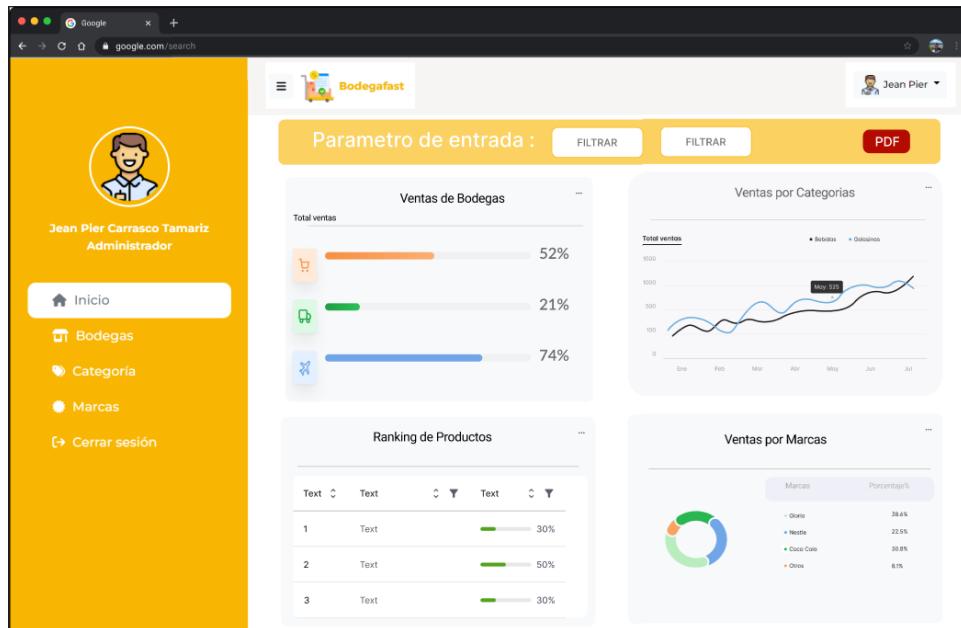
6.1. Primer diseño

Figura 124. Primer diseño de las alternativas de Dashboard



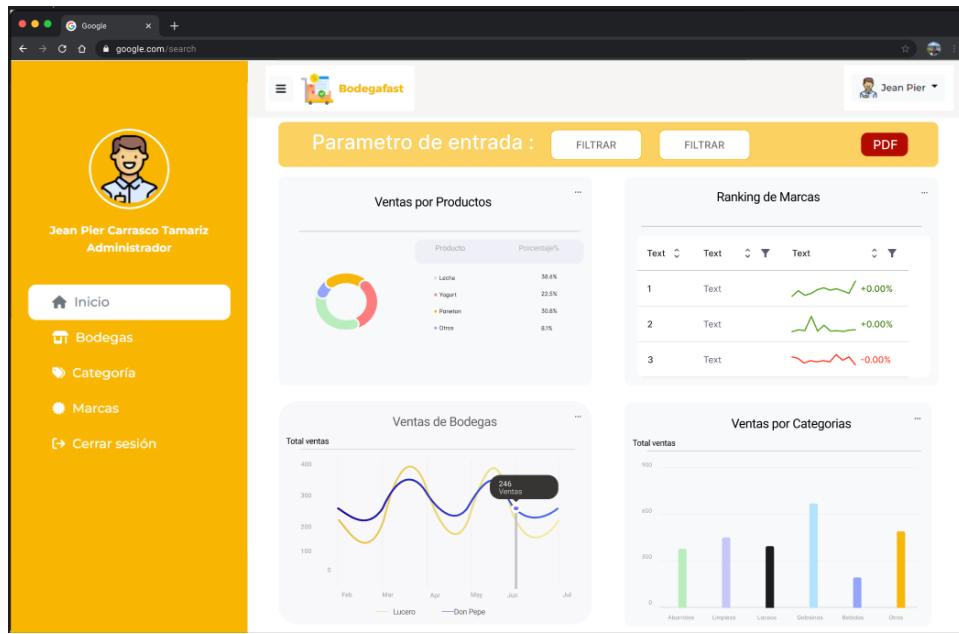
6.2. Segundo diseño

Figura 125. Segundo diseño de las alternativas de Dashboard



6.3. Tercer diseño

Figura 126. Tercer diseño de las alternativas de Dashboard



6.4. Selección de tablero

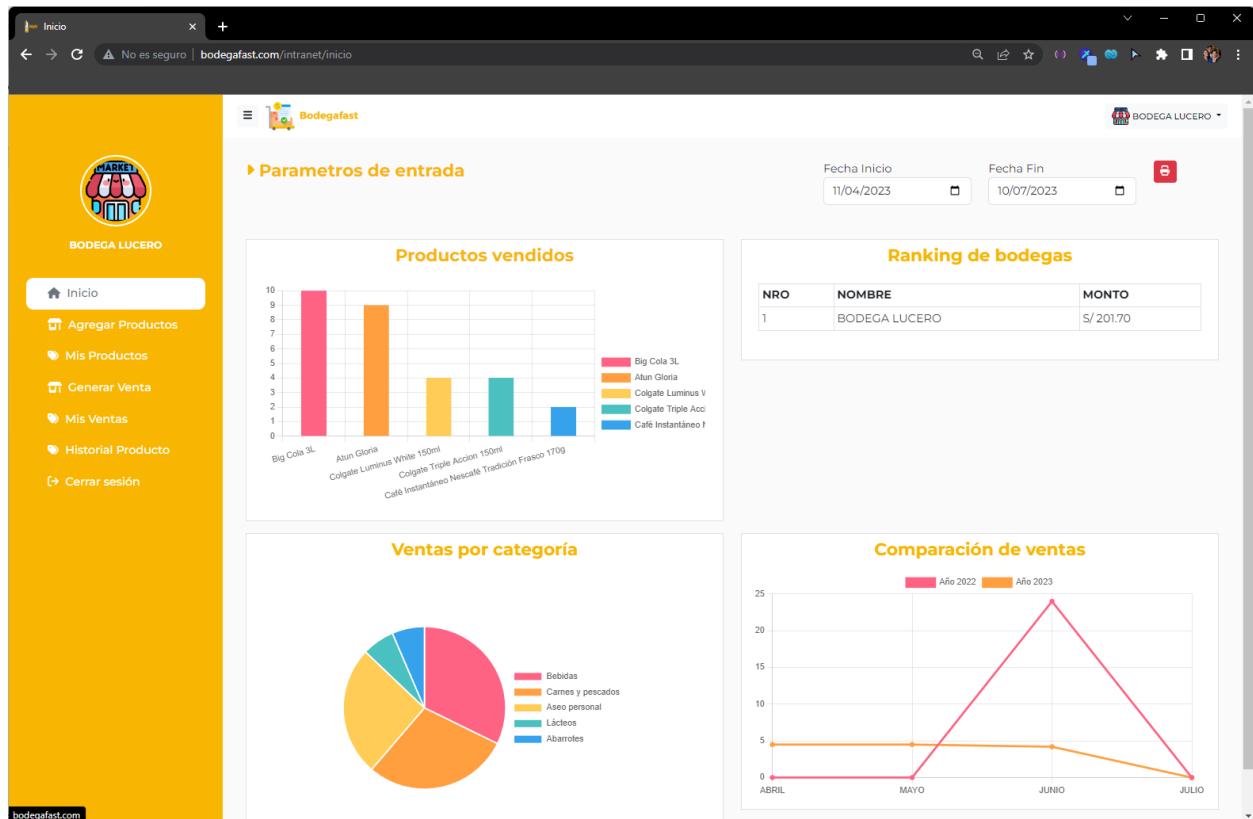
Tabla 7. Tabla de selección de las alternativas de Dashboard

Factor	Peso	Diseño 1		Diseño 2		Diseño 3	
		Nota	Puntaje	Nota	Puntaje	Nota	Puntaje
Disponibilidad de datos	20%	4	0.8	3	0.6	4	0.8
Toma de decisiones	30%	5	1.5	4	1.2	4	1.2
Facilidad de usos	30%	5	1.5	4	1.2	3	0.9

Modificabilidad	20%	4	0.8	5	1	4	0.8
Total	4.6		4		3.7		

6.5. Diseño del dashboard para bodegas – Sistema Bodegafast

Figura 127. Sistema de Bodegafast



Para la creación de gráficos estadísticos se utilizó la librería Chart Js

Creación del gráfico de barras – productos vendidos

Figura 128. Código de creación del grafico de barras

```

let datosVentas = {
    labels: ['Leche Gloria', 'Cafe Instantaneo', 'Leche Condensada', 'Aceite Vegetal',
    'Big Cola'],
    datasets: [{

        label : 'Productos',
        barPercentage: 0.5,
        data: [12, 19, 3, 5, 2],
        backgroundColor: listaColores,
        borderColor: listaBordes,
        borderWidth: 1
    }]
};

let myChartVentas = new Chart(ctxVentasProductos.getContext('2d'), {
    type: 'bar',
    data: datosVentas,
    options: {
        responsive:true,
        plugins : {
            legend : { ... }
        }
    }
});

```

Creación del gráfico del pastel – ventas por categoría

Figura 129. Código de creación del grafico del pastel

```

let datosCategorias = [
    labels:['Leche Gloria', 'Cafe Instantaneo', 'Leche Condensada', 'Aceite Vegetal',
    'Big Cola'],
    datasets: [{

        data: [300, 50, 100,60,80],
        backgroundColor: listaColores,
        hoverOffset: 1,
    }]
];

let myChartVentasCategoria = new Chart(ctxProductosCategoria.getContext('2d'), {
    type: 'pie',
    data: datosCategorias,
    options: {
        responsive:true,
        plugins : {
            legend : {
                position : 'right'
            }
        },
    }
});
let myChartComparacionVentas = new Chart(ctxComparacionVentas.getContext('2d')) {

```

Creación del gráfico de líneas – comparación de ventas

Figura 130. Código de creación del grafico de lineas

```
let myChartComparacionVentas = new Chart(ctxComparacionVentas.getContext('2d'), {
    type:'line',
    data : [
        labels : ['Abril','Mayo','Junio','Julio'],
        datasets: [
            {
                label : 'Año 2022',
                data: [300, 50, 100,60,80],
                backgroundColor : listaColores[0],
                borderColor : listaColores[0]
            },
            {
                label : 'Año 2023',
                data: [100, 20, 80,150,200],
                backgroundColor : listaColores[1],
                borderColor : listaColores[1]
            }
        ],
    ]
})
```

Inserción de datos traídos desde el backend para incorporarlos en los gráficos

Estructura de la capa vista del dashboard

Figura 131. Código de estructura de capa vista del dashboard 1

```
async function cargarDashboard(){
    if(!txtFechaInicio.value || !txtFechaFin.value){
        return helper.alertaToast("error","Estableza los parametros de fecha correctas");
    }
    let datos = new FormData();
    datos.append("fInicio",txtFechaInicio.value);
    datos.append("fFin",txtFechaFin.value);
    datos.append("acciones",'solicitar-datos');
    try {
        const response = await fetch(helper.urlDashboardBodega,{
            method:"POST",
            body : datos
        }).then(result => result.json());
        //LLENADO DE DATOS PARA EL GRAFICO LINEAL
        myChartComparacionVentas.data.labels = response.ventasRealizadas.year1.map(mes => mes.nombreMes);
        myChartComparacionVentas.data.datasets[1].data = response.ventasRealizadas.year1.map(mes => mes.montoVendido);
        myChartComparacionVentas.data.datasets[0].data = response.ventasRealizadas.year2.map(mes => mes.montoVendido);
        myChartComparacionVentas.update();
        //LLENADO DE DATOS PARA EL GRAFICO DEL PASTEL
        myChartVentasCategoria.data.labels = response.productosVendidosCategoria.map(categoría => categoría.nombreCategoria);
        myChartVentasCategoria.data.datasets[0].data = response.productosVendidosCategoria.map(categoría => categoría.cantidadVendida);
        myChartVentasCategoria.update();
        //LLENADO DE DATOS PARA EL GRAFICO DE BARRAS
        myChartVentas.data.labels = response.productosVendidos.map(producto => producto.nombre);
        myChartVentas.data.datasets[0].data = response.productosVendidos.map(producto => producto.cantidadVendida);
        //LLENADO DE DATOS PARA EL GRAFICO DE PASTEL
        myChartVentasPastel.data.labels = response.productosVendidos.map(producto => producto.nombre);
        myChartVentasPastel.data.datasets[0].data = response.productosVendidos.map(producto => producto.cantidadVendida);
        myChartVentasPastel.update();
    } catch (error) {
        helper.alertaToast("error",error.message);
    }
}
```

Figura 132. Código de estructura de capa vista del dashboard 2

```

<section class="P-2">
    <div class="row mb-5 filtros-cabecera">
        <h4 class="col-12 col-lg-6 col-xl-7 titulo-principal-modulo">
            <i class="fa-solid fa-caret-right"></i> Parametros de entrada
        </h4>
        <div class="col-6 col-lg-3 col-xl-2">
            <label for="txtFechaInicio">Fecha Inicio</label>
            <input type="date" class="form-control filtro-aplicar" value="<?= date('Y-m-d', strtotime("-90 days", strtotime(date('Y-m-d')))) ?>" id="txtFechaInicio">
        </div>
        <div class="col-6 col-lg-3 col-xl-2">
            <label for="txtFechaFin">Fecha Fin</label>
            <input type="date" id="txtFechaFin" class="form-control filtro-aplicar" value="<?= date('Y-m-d') ?>">
        </div>
        <div class="col-6 col-lg-3 col-xl-1">
            <button type="button" class="btn btn-sm btn-danger" title="Imprimir" id="btnImprimirPdf"><i class="fas fa-print"></i></button>
        </div>
    </div>
    <div class="row">
        <div class="graficos-informes mb-3 col-12 col-lg-6">
            <div class="border p-2 bg-white">
                <h4 class="text-center titulo-principal-modulo">Productos vendidos</h4>
                <div class="p-3">
                    <canvas id="productosVendidos"></canvas>
                </div>
            </div>
        </div>
        <div class="graficos-informes mb-3 col-12 col-lg-6">...
        </div>
        <div class="graficos-informes mb-3 col-12 col-lg-6">...
        </div>
        <div class="graficos-informes mb-3 col-12 col-lg-6">...
        </div>
    </div>
</section>

```

Estructura de la capa controlador del dashboard

Figura 133. Código de estructura de capa controlador del dashboard

```

<?php
//Establecemos el nombre de espacio donde se trabaja
namespace Controllers\Bodega;
//Requerimos los archivos necesarios para el funcionamiento
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/CategoríaModel.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/UsuarioModel.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/BodegaModel.php';
require_once $_SERVER['DOCUMENT_ROOT'] . '/Models/ProductoModel.php';

//Usamos las clases y lo renombramos

use Models\Categoría as CategoríaModel;
use Models\Usuario as UsuarioModel;
use Models\Bodega as BodegaModel;
use Models\Producto as ProductoModel;

class Dashboard {

    public function obtenerProductosMasVendidos(string $fechaInicio, string $fechaFin)
    {
        $usuarioModel = new UsuarioModel();
        $data = $usuarioModel→obtenerDatosAutenticado();
        if (empty($data)) {
            return ['session' => true];
        }
        if (!in_array($data['rol'], [$usuarioModel→rolBodega])) {
            return ['session' => true];
        }
        $productoModel = new ProductoModel();
        $productoModel→setIdBodega($data['idAccesoRol']);
        return $productoModel→obtenerProductosDashboard($fechaInicio,$fechaFin);
    }

    public function obtenerProductosMasCategoría(string $fechaInicio, string $fechaFin)
    {
        $usuarioModel = new UsuarioModel();
        $data = $usuarioModel→obtenerDatosAutenticado();
        if (empty($data)) {
            return ['session' => true];
        }
        if (!in_array($data['rol'], [$usuarioModel→rolBodega])) {
            return ['session' => true];
        }
        $categoríaModel = new CategoríaModel();
        return $categoríaModel→obtenerProductosCategoríaDashboard($fechaInicio,$fechaFin,$data['idAccesoRol']);
    }

    public function obtenerRanking(string $fechaInicio, string $fechaFin) : array
    {
        $usuarioModel = new UsuarioModel();
        $data = $usuarioModel→obtenerDatosAutenticado();
        if (empty($data)) {
            return ['session' => true];
        }
        if (!in_array($data['rol'], [$usuarioModel→rolBodega])) {
            return ['session' => true];
        }
        $bodegaModel = new BodegaModel();
        return $bodegaModel→rankingDashboard($fechaInicio,$fechaFin);
    }

    public function obtenerVentas($fechaInicio,$fechaFin)
    {
        $usuarioModel = new UsuarioModel();
        $data = $usuarioModel→obtenerDatosAutenticado();
        if (empty($data)) {
            return ['session' => true];
        }
        if (!in_array($data['rol'], [$usuarioModel→rolBodega])) {
            return ['session' => true];
        }
        $productoModel = new ProductoModel();
        $productoModel→setIdBodega($data['idAccesoRol']);
        return $productoModel→obtenerVentasYearDashboard($fechaInicio,$fechaFin);
    }
}

```

Estructura de la capa modelo del dashboard

Figura 134. Código de estructura de capa modelo del dashboard

```

public function rankingDashboard(string $fechaInicio, string $fechaFin){
    $cn = $this->conectar();
    $stmt = $cn->prepare("CALL SP_DASHBOARD_RANKIN_BODEGAS(?,?)");
    $stmt->bind_param("ss", $fechaInicio, $fechaFin);
    $stmt->execute();
    $rs = $stmt->get_result();
    $result = [];
    while ($result[] = $rs->fetch_assoc());
    array_pop($result);
    $stmt->close();
    return $result;
}

public function obtenerProductosDashboard(string $fechaInicio, string $fechaFin) {
    $cn = $this->conectar();
    $stmt = $cn->prepare("CALL SP_DASHBOARD_PRODUCTOS_VENDIDOS(?,?,?,?)");
    $stmt->bind_param("ssi", $fechaInicio, $fechaFin,$this->idBodega);
    $stmt->execute();
    $rs = $stmt->get_result();
    $result = [];
    while ($result[] = $rs->fetch_assoc());
    array_pop($result);
    $stmt->close();
    return $result;
}

public function obtenerVentasYearDashboard(string $fechaInicio, string $fechaFin) {
    $cn = $this->conectar();
    $stmt = $cn->prepare("CALL SP_DASHBOARD_PRODUCTOS_VENDIDOS_YEAR(?,?,?,?)");
    $stmt->bind_param("ssi", $fechaInicio, $fechaFin,$this->idBodega);
    $stmt->execute();
    $rs = $stmt->get_result();
    $result = [];
    while ($result[] = $rs->fetch_assoc());
    array_pop($result);
    $stmt->close();
    return $result;
}

public function obtenerProductosCategoriaDashboard(string $fechaInicio, string $fechaFin, int $idBodega)
{
    $cn = $this->conectar();
    $stmt = $cn->prepare("CALL SP_DASHBOARD_PRODUCTOS_VENDIDOS_CATEGORIA(?,?,?,?)");
    $stmt->bind_param("ssi", $fechaInicio, $fechaFin,$idBodega);
    $stmt->execute();
    $rs = $stmt->get_result();
    $result = [];
    while ($result[] = $rs->fetch_assoc());
    array_pop($result);
    $stmt->close();
    return $result;
}

```

Estructura del procedimiento almacenado - Base de datos del dashboard

Figura 135. Código de estructura del procedimiento almacenado (parte 1)

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `SP_DASHBOARD_PRODUCTOS_VENDIDOS`(
    IN _fechaInicio DATE,
    IN _fechaFin DATE,
    IN _idBodega INT)
BEGIN
    SELECT
        p.nombre,
        SUM( v.cantidad ) AS cantidadVendida
    FROM
        productos p
    INNER JOIN (
        SELECT
            vd.*
        FROM
            ventas_detalle vd
        INNER JOIN ventas ve ON ve.id = vd.id_venta
        WHERE
            DATE_FORMAT( ve.fecha_venta, '%Y-%m-%d' ) BETWEEN _fechaInicio
            AND _fechaFin
    ) AS v ON p.id = v.id_producto
    INNER JOIN bodegas b ON p.id_bodega = b.id
    WHERE
        b.id = _idBodega
    GROUP BY
        p.id
    ORDER BY
        cantidadVendida DESC
    LIMIT 5;
END
```

Figura 136. Código de estructura del procedimiento almacenado (parte 2)

```
CREATE DEFINER='root'@'localhost' PROCEDURE `SP_DASHBOARD_RANKIN_BODEGAS`(
    IN _fechaInicio DATE,
    IN _fechaFin DATE)
BEGIN
    SELECT
        b.nombre AS nombreBodega,
        SUM( v.subtotal ) AS montoVendido
    FROM
        bodegas b
    INNER JOIN (
        SELECT
            vd.subtotal,
            p.id_bodega
        FROM
            ventas_detalle vd
        INNER JOIN productos p ON vd.id_producto = p.id
        INNER JOIN ventas ve ON ve.id = vd.id_venta
        WHERE
            DATE_FORMAT( ve.fecha_venta, '%Y-%m-%d' ) BETWEEN _fechaInicio
            AND _fechaFin
    ) AS v ON b.id = v.id_bodega
    GROUP BY
        b.id
    ORDER BY
        montoVendido DESC
    LIMIT 5;
END
```

Figura 137. Código de estructura del procedimiento almacenado (parte 3)

```

CREATE DEFINER=`root`@`localhost` PROCEDURE `SP_DASHBOARD_PRODUCTOS_VENDIDOS_CATEGORIA`(
    IN _fechaInicio DATE,
    IN _fechaFin DATE,
    IN _idBodega INT)
BEGIN
    SELECT
        pc.nombreCategoria,
        SUM( v.cantidad ) AS cantidadVendida
    FROM
        productos p
    INNER JOIN (
        SELECT
            vd.*
        FROM
            ventas_detalle vd
        INNER JOIN ventas ve ON ve.id = vd.id_venta
        WHERE
            DATE_FORMAT( ve.fecha_venta, '%Y-%m-%d' ) BETWEEN _fechaInicio
            AND _fechaFin
    ) AS v ON p.id = v.id_producto
    INNER JOIN bodegas b ON p.id_bodega = b.id
    INNER JOIN (
        SELECT
            c.id AS idCategoria,
            c.nombre AS nombreCategoria,
            pc.id_producto
        FROM
            categorias c
        INNER JOIN productos_categorias pc ON c.id = pc.id_categoria
    ) pc ON pc.id_producto = p.id
    WHERE
        b.id = _idBodega
    GROUP BY
        pc.idCategoria
    ORDER BY
        cantidadVendida DESC
    LIMIT 5;
END

```

Figura 138. Código de estructura del procedimiento almacenado (parte 4)

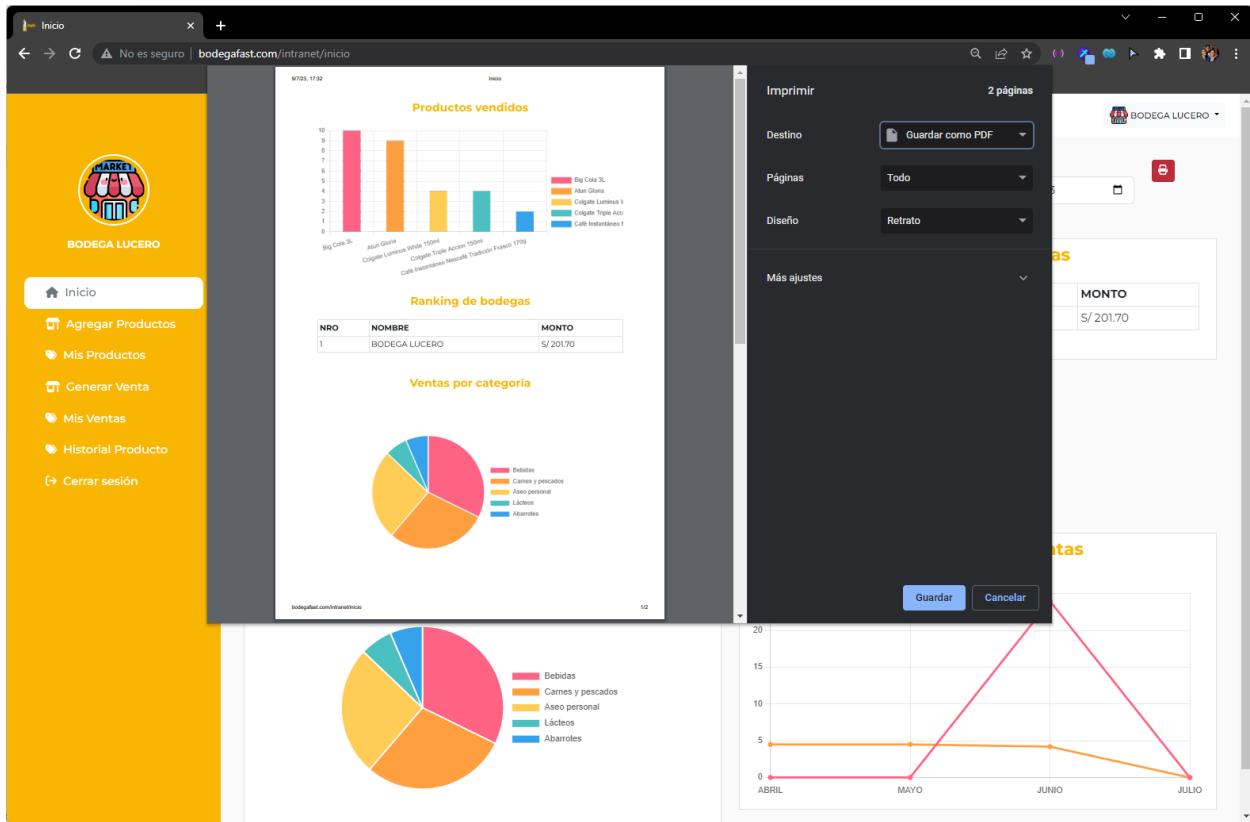
```

CREATE DEFINER='root'@'localhost' PROCEDURE
`SP_DASHBOARD_PRODUCTOS_VENDIDOS_YEAR`(
IN _fechaInicio DATE,
IN _fechaFin DATE,
IN _idBodega INT)
BEGIN
SET lc_time_names = 'es_ES';
SET sql_mode = '';
WITH RECURSIVE mes ( n ) AS ( SELECT MONTH ( _fechaInicio ) UNION ALL SELECT n
+ 1 FROM mes WHERE n < MONTH ( _fechaFin ) ) SELECT
    UPPER(
        MONTHNAME(
            CONCAT( '2022-', LPAD( n, 2, 0 ), '-01' ))) AS nombreMes,
SELECT
    IF
        ( SUM( vd.subtotal ) IS NULL, 0, vd.subtotal )
    FROM
        ventas_detalle vd
        INNER JOIN ( SELECT p.* FROM productos p INNER JOIN bodegas b ON p.
id_bodega = b.id WHERE b.id = _idBodega ) AS b ON b.id = vd.id_producto
        INNER JOIN ventas v ON v.id = vd.id_venta
    WHERE
        DATE_FORMAT( v.fecha_venta, '%Y-%m-%d' ) BETWEEN _fechaInicio
        AND _fechaFin
        AND MONTH ( fecha_venta ) = n
    ) AS montoVendido
FROM
    mes;
END

```

6.6. Exportar en pdf – dashboard

Figura 139. Exportación en PDF del dashboard



Actualmente los navegadores ya incorporan las opciones de guardar en pdf al momento de imprimir, el cual al presionar el botón **guardar** se tendrá que seleccionar la ruta en donde el archivo pdf será almacenado, para darle estilos al documento para imprimir se utilizó un archivo css el cual lo adaptamos para que se adapte a una hoja A4.

6.7. Exportación de ventas en formato Excel

Figura 140. Exportación de las ventas en formato Excel

Nº Venta	Fecha	Cliente	Celular	Dirección	Nº Productos	Cantidad	Importe	I.G.V	Total	Acciones
00024	14/06/2023	Yosimar Ramirez	987845123	Av. Los libertadores 14 - Carabayllo	2	2	\$/ 5.25	\$/ 1.15	\$/ 6.40	Visualizar Anular
00027	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	1	\$/ 3.69	\$/ 0.81	\$/ 4.50	Visualizar Anular
00029	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	1	\$/ 3.69	\$/ 0.81	\$/ 4.50	Visualizar Anular
00030	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	1	\$/ 3.69	\$/ 0.81	\$/ 4.50	Visualizar Anular
00033	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	2	\$/ 13.12	\$/ 2.88	\$/ 16.00	Visualizar Anular
00034	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	1	\$/ 6.56	\$/ 1.44	\$/ 8.00	Visualizar Anular
00035	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	1	\$/ 6.56	\$/ 1.44	\$/ 8.00	Visualizar Anular
00036	14/06/2023	Osbaldo Laurencio	984545451	Av. los jasmines 351 - Comas	1	1	\$/ 3.69	\$/ 0.81	\$/ 4.50	Visualizar Anular
00037	14/06/2023	Osbaldo	984545451	Av. los jasmines 351 - Comas	2	2	\$/ 10.25	\$/ 2.25	\$/ 12.50	Visualizar Anular

Para exportar los datos Excel debemos ingresar como un usuario de bodega e ingresar al módulo de mis ventas, se define las fechas en los filtros, por defecto están los valores de los intervalos de 30 días, después presionar el botón de reportes, seleccionar Excel – detalle de venta y se descargará el reporte

Figura 141. Descarga desde el navegador el reporte en excel



Figura 142. Reporte en excel del detalle de ventas

Nº Venta	Fecha Venta	Cliente	Método Pago	Producto	Precio	Cantidad	Subtotal
24	14/06/2023	Yosimar Ramirez	Efectivo	Leche Gloria Azul de 400g	\$/ 4.20	1	\$/ 4.20
24	14/06/2023	Yosimar Ramirez	Efectivo	Leche Gloria Azul de 170g	\$/ 2.20	1	\$/ 2.20
27	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
29	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
30	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
33	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	2	\$/ 16.00
34	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	1	\$/ 8.00
35	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	1	\$/ 8.00
36	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
37	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
37	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	1	\$/ 8.00
38	14/06/2023	Osbaldo Laurencio	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
38	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	2	\$/ 16.00
39	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	1	\$/ 8.00
40	14/06/2023	Osbaldo Laurencio	CARD	Big Cola 3L	\$/ 8.00	2	\$/ 16.00
41	14/06/2023	Osbaldo Laurencio	CARD	Colgate Luminus White 150ml	\$/ 6.70	3	\$/ 20.10
42	14/06/2023	Osbaldo Laurencio	CARD	Colgate Triple Accion 150ml	\$/ 4.50	1	\$/ 4.50
43	14/06/2023	Osbaldo Laurencio	CARD	Colgate Triple Accion 150ml	\$/ 4.50	2	\$/ 9.00
44	14/06/2023	Luis Alcantara	CARD	Café Instantáneo Nescafé Tradición Frasco 170g	\$/ 15.00	1	\$/ 15.00
44	14/06/2023	Luis Alcantara	CARD	Colgate Triple Accion 150ml	\$/ 4.50	1	\$/ 4.50
45	14/06/2023	Luis Alcantara	CARD	Atun Gloria	\$/ 4.50	1	\$/ 4.50
46	16/06/2023	Osbaldo Laurencio	CARD	Café Instantáneo Nescafé Tradición Frasco 170g	\$/ 15.00	1	\$/ 15.00
46	16/06/2023	Osbaldo Laurencio	CARD	Colgate Luminus White 150ml	\$/ 6.70	1	\$/ 6.70
25			Total		\$/ 150.80	29	\$/ 192.70

Figura 143. Controlador modificado para la exportación en excel

```
//se incluye una vista html
if($_POST['accion'] == "pdf"){
    ob_start();
    include_once $_SERVER['DOCUMENT_ROOT'] . '/Views/Bodega/reportes/detalleVenta.php';
    $html = ob_get_clean();
    $dompdf = new Dompdf();
    //Obtencion de la vista
    $dompdf->loadHtml($html);
    //definimos el papel horizontal
    $dompdf->setPaper('A4', 'landscape');
    //renderizamos en el navegador
    $dompdf->render();
    $dompdf->stream("reporte_ventas.pdf",array("Attachment" => false));
} else{
    header("Content-Type: application/xls");
    header('Content-Type: text/html; charset=utf-8');
    header("Content-Disposition: attachment; filename=reporte_de_ventas_" . date('Y:m:d:m:s') . ".xls");
    header("Pragma: no-cache");
    header("Expires: 0");
    ob_start();
    include_once $_SERVER['DOCUMENT_ROOT'] . '/Views/Bodega/reportes/detalleVentaExcel.php';
    echo ob_get_clean();
}
```

Estructura del modelo para obtener los datos del reporte

Figura 144. Código de estructura del modelo para obtener los datos del reporte

```
public function verVentasPorBodega(int $idBodega, string $fechaInicio, string $fechaFin)
{
    $cn = $this->conectar();
    $stmt = $cn->prepare("CALL SP_R_T_VENTAS_BODEGAS(?, ?, ?)");
    $stmt->bind_param("iss", $idBodega, $fechaInicio, $fechaFin);
    $stmt->execute();
    $rs = $stmt->get_result();
    $result = [];
    while ($result[] = $rs->fetch_assoc());
    array_pop($result);
    $stmt->close();
    return $result;
}
```

Estructura del procedimiento almacenado – base de datos

Figura 145. Código de estructura del procedimiento de almacenado

```
1 CREATE DEFINER=`root`@`localhost` PROCEDURE `SP_R_T_VENTAS_BODEGAS`(
2     IN _bodega INT,
3     IN _fechaInicio DATE,
4     IN _fechaFin DATE
5 )
6 BEGIN
7     SELECT
8         v.id,
9         DATE_FORMAT(v.fecha_venta, '%d/%m/%Y') AS fecha,
10        LPAD(v.id, 5, '0') AS nroVenta,
11        CONCAT(u.nombres, ' ', u.apellidos) AS nombresCliente,
12        v.celular,
13        v.metodo_pago,
14        v.direccion,
15        SUM(vd.cantidad) AS cantidadProductos,
16        COUNT(vd.id) AS nroProductos,
17        ROUND((SUM(vdsubtotal) - SUM(vdsubtotal) * 0.18), 2) AS subtotal,
18        ROUND((SUM(vdsubtotal) * 0.18), 2) AS igv,
19        SUM(vdsubtotal) AS total
20    FROM
21        ventas_detalle vd
22        INNER JOIN ventas v ON vd.id_venta = v.id
23        INNER JOIN productos p ON vd.id_producto = p.id
24        INNER JOIN usuarios u ON u.id = v.id_usuarios
25    WHERE
26        v.estado != 0
27        AND p.id_bodega = _bodega
28        AND DATE_FORMAT(v.fecha_venta, '%Y-%m-%d') BETWEEN _fechaInicio
29        AND _fechaFin
30    GROUP BY
31        v.id;
32 END
```

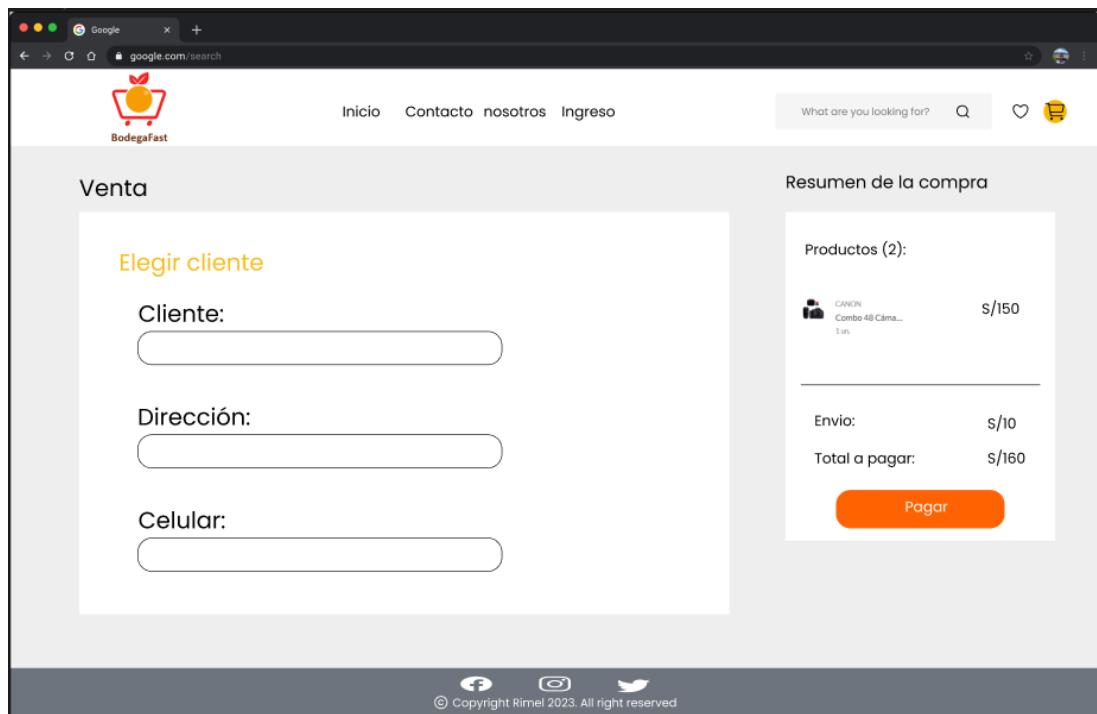
Estructura de la vista para exportar datos en Excel

Figura 146. Código de la vista para exportar datos en Excel

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5  </head>
6  <body>
7  <table border="1">
8      <thead>
9          <tr>
10             <th>Nº Venta</th>
11             <th>Fecha Venta</th>
12             <th style="width: 200px;">Cliente</th>
13             <th>Método Pago</th>
14             <th style="width: 300px;">Producto</th>
15             <th>Precio</th>
16             <th>Cantidad</th>
17             <th>Subtotal</th>
18     </tr>
19  </thead>
20  <tbody>
21      <?php
22          $precio = 0;
23          $cantidad = 0;
24          $subtotal = 0;
25          foreach ($ventas as $k=>$venta) {
26              foreach ($venta['productos'] as $pk=>$p) {
27                  $precio += $p['precio_venta'];
28                  $cantidad += $p['cantidad'];
29                  $subtotal += $p['subtotal'];
30                  echo "<tr>
31                      <td>".$venta['nroVenta']."</td>
32                      <td>".$venta['fecha']."</td>
33                      <td>".$venta['nombresCliente']."</td>
34                      <td>".$venta['metodo_pago']."</td>
35                      <td>".$p['nombre']."'".</td>
36                      <td>S/" . $p['precio_venta'] . "</td>
37                      <td class='text-center'>". $p['cantidad'] ."</td>
38                      <td>S/" . $p['subtotal'] . "</td></tr>
39                  ";
40              }
41          }
42      ?>
43  </tbody>
```

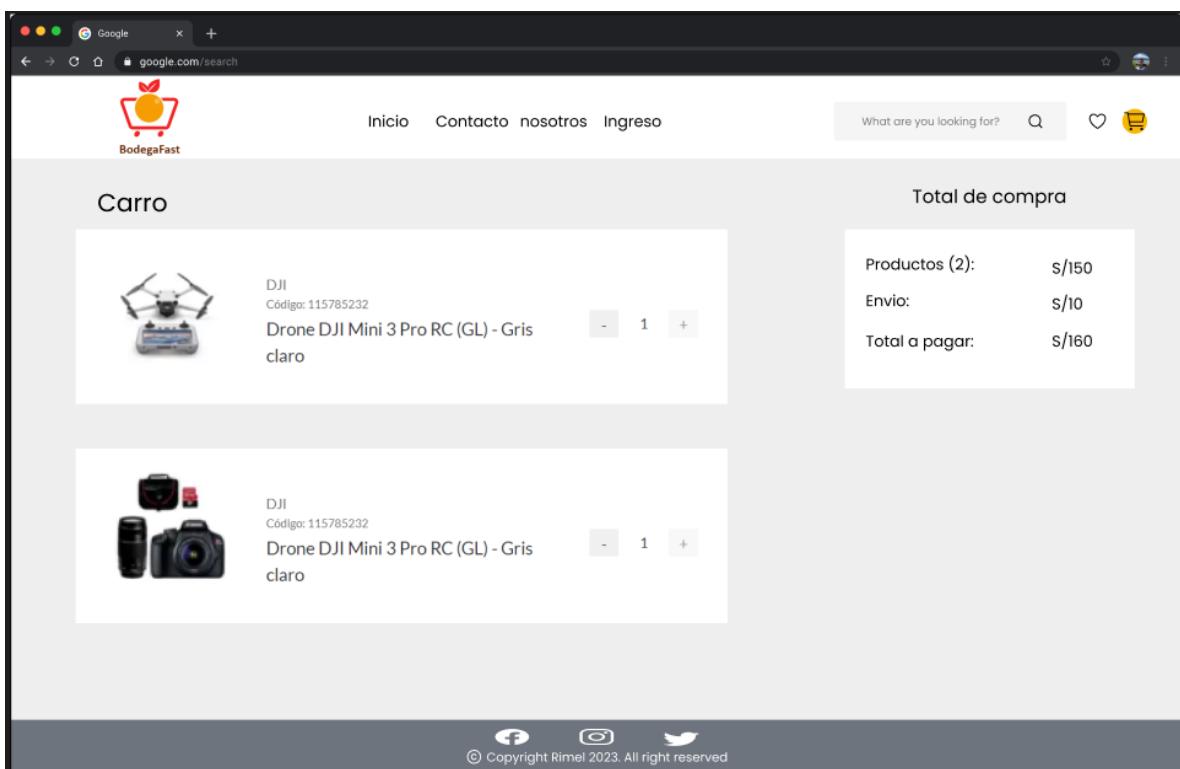
6.8. Mockups adicionales

Figura 147. Mockup de Venta bodega



Mockup - carrito de compra

Figura 148. Mockup de Carrito de compra



Mockup - métodos de pago

Figura 149. Mockup de métodos de pago (parte 1)

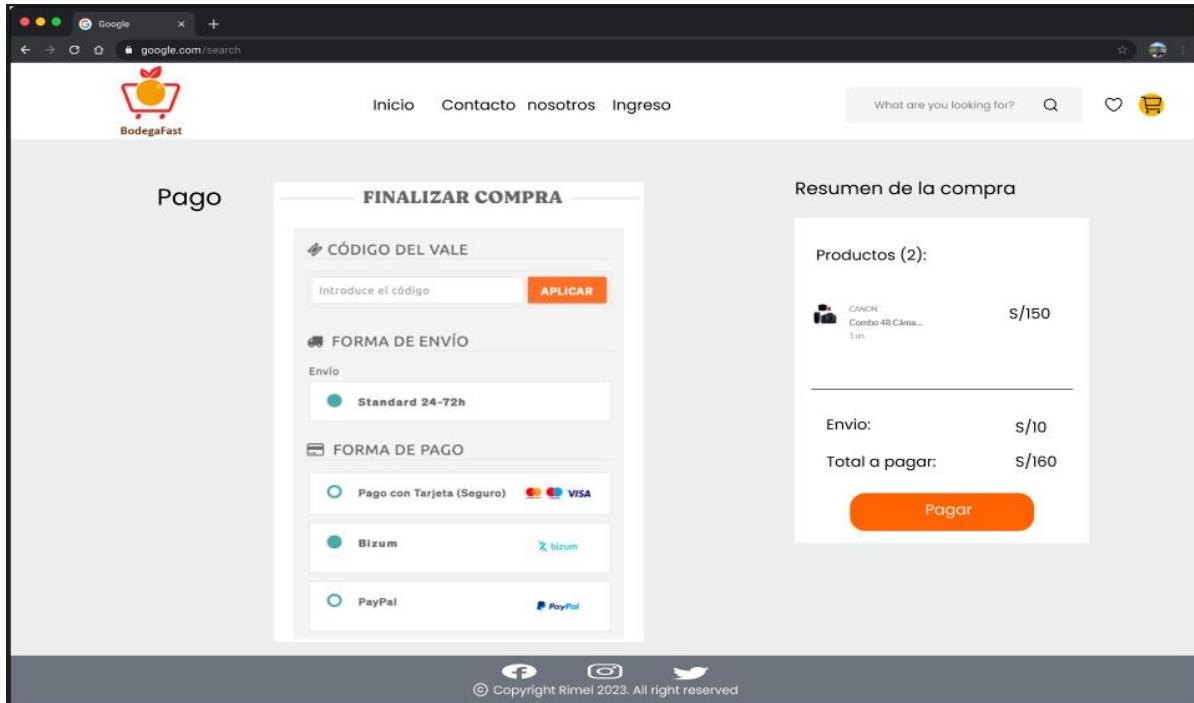
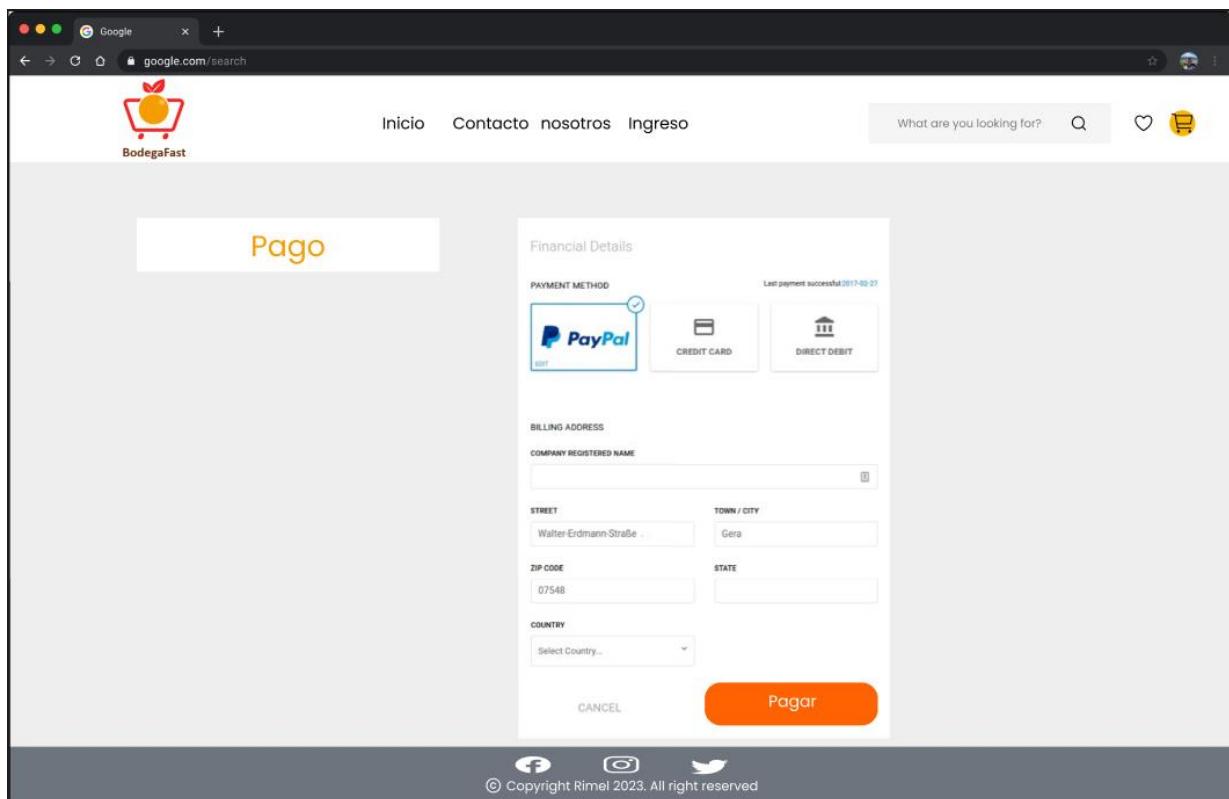
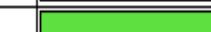


Figura 150. Mockup de Carrito de compra (parte 2)



6.9. Evaluación de la cobertura de código mediante PHPUNIT

Figura 151. Tabla de resultados arrojados por PHPUNIT

File		Statements		Branches		Function		Lines	
Controllers/		100.00%	1/1	100.00%	1/1	100.00%	1/1	100.00%	1/1
Controllers/Administrador/Bodegas.php		93.75%	71/78	88.75%	71/78	100.00%	78/78	100.00%	28/28
Controllers/Administrador/Categorias.php		95.80%	75/79	100.00%	45/45	100.00%	55/55	100.00%	71/71
Controllers/Administrador/Marcas.php		100.00%	80/80	100.00%	80/80	100.00%	66/66	100.00%	80/80
Controllers/Bodega/Productos.php		95.83%	76/81	89.55%	71/77	100.00%	45/45	100.00%	75/75
Controllers/Bodega/Venta.php		94.12%	76/82	91.33%	79/86	100.00%	22/22	100.00%	82/82
Controllers/Cliente/Compras.php		100.00%	71/71	100.00%	82/82	100.00%	41/41	100.00%	71/71
Controllers/Correo.php		97.14%	71/75	97.14%	71/74	100.00%	22/22	100.00%	74/74
Controllers/Login.php		93.00%	79/85	93.00%	71/76	100.00%	37/37	100.00%	66/66
Controllers/PaginaPrincipal.php		95.94%	71/86	95.94%	52/55	100.00%	52/52	100.00%	58/58
Http/Administrador/Bodega.php		100.00%	88/88	97.00%	83/86	100.00%	60/60	100.00%	67/67
Http/Administrador/Categoría.php		100.00%	75/75	97.00%	83/86	100.00%	71/71	100.00%	71/71
Http/Administrador/Marca.php		95.45%	83/89	97.00%	85/89	100.00%	28/28	100.00%	28/28
Http/Bodega/Productos.php		99.00%	88/90	97.00%	41/43	100.00%	39/39	100.00%	39/39
Http/Bodega/Ventas.php		92.85%	84/91	97.00%	89/91	100.00%	22/20	100.00%	22/20
Http/Cliente/Compras.php		100.00%	82/82	97.00%	71/74	100.00%	22/21	100.00%	22/21
Http/Usuario.php		97.55%	78/83	97.00%	71/73	100.00%	22/22	100.00%	22/22