

Desarrollo de software

Documento final

Komalli

Equipo 4
28/05/2024

Índice

Índice	2
Introducción	11
Antecedentes	11
Contexto y Situación Actual	11
Justificación	11
Objetivos	12
Métodología	12
Estructura del Trabajo	14
1 Requerimientos	15
1.1 Alcance del Proyecto	15
1.2 Clases y Características de usuarios	15
1.3 Requerimientos Funcionales	17
1.4 Requerimientos No Funcionales	18
1.5 Modelo de Casos de Uso	19
2 Administración del Proyecto	20
2.1 Lista de Casos de Uso	20
2.2 Lista de tareas	21
3 Análisis y Diseño	25
3.1 CU 01 Registrar Usuario	25
3.1.1 Prototipo	25
3.1.2 Descripción	26
3.1.3 Diagrama de robustez	27
3.1.4 Diagrama de secuencia	28
3.2 CU 02 Modificar Usuario	29
3.2.1 Prototipo	29
3.2.2 Descripción	30
3.2.3 Diagrama de robustez	31
3.2.4 Diagrama de secuencia	32
3.3 CU 03 Levantar pedido	33
3.3.1 Prototipo	33
3.3.2 Descripción	34
3.3.3 Diagrama de robustez	36
3.3.4 Diagrama de secuencia	37

3.4 CU 04 Consultar usuario	38
3.4.1 Prototipo	38
3.4.2 Descripción	38
3.4.3 Diagrama de robustez	39
3.4.4 Diagrama de secuencia	40
3.5 CU 05 Agregar ingrediente a inventario	40
3.5.1 Prototipo	40
3.5.2 Descripción	42
3.5.3 Diagrama de robustez	43
3.5.4 Diagrama de secuencia	44
3.6 CU 06 Modificar ingrediente	45
3.6.1 Prototipo	45
3.6.2 Descripción	46
3.6.3 Diagrama de robustez	47
3.6.4 Diagrama de secuencia	48
3.7 CU 07 Buscar ingrediente en inventario	49
3.7.1 Prototipo	49
3.7.2 Descripción	51
3.7.3 Diagrama de robustez	52
3.7.4 Diagrama de secuencia	52
3.8 CU 08 Generar reporte de Inventario	53
3.8.1 Prototipo	53
3.8.2 Descripción	54
3.8.3 Diagrama de robustez	55
3.8.4 Diagrama de secuencia	56
3.9 CU 09 Genera pedido	57
3.9.1 Prototipo	57
3.9.2 Descripción	59
3.9.3 Diagrama de robustez	60
3.9.4 Diagrama de secuencia	61
3.10 CU 10 Editar pedido	62
3.10.1 Prototipo	62
3.10.2 Descripción	63
3.10.3 Diagrama de robustez	64
3.10.4 Diagrama de secuencia	65
3.11 CU 11 Editar Estado de pedido	66
3.11.1 Prototipo	66
3.11.2 Descripción	67
3.11.3 Diagrama de robustez	68
3.11.4 Diagrama de secuencia	68
3.12 CU 12 Publicar Menu	69
3.12.1 Prototipo	69

3.12.2	Descripción _____	71
3.12.3	Diagrama de robustez _____	72
3.12.4	Diagrama de secuencia _____	73
3.13	CU 13 Editar Menu _____	74
3.13.1	Prototipo _____	74
3.13.2	Descripción _____	76
3.13.3	Diagrama de robustez _____	77
3.13.4	Diagrama de secuencia _____	78
3.14	CU 14 Eliminar Menu _____	79
3.14.1	Prototipo _____	79
3.14.2	Descripción _____	80
3.14.3	Diagrama de robustez _____	82
3.14.4	Diagrama de secuencia _____	83
3.15	CU 15 Consultar Menu _____	84
3.15.1	Prototipo _____	84
3.15.2	Descripción _____	85
3.15.3	Diagrama de robustez _____	86
3.15.4	Diagrama de secuencia _____	87
3.16	CU 16 Agregar Comentario a bitácora _____	88
3.16.1	Prototipo _____	88
3.16.2	Descripción _____	90
3.16.3	Diagrama de robustez _____	91
3.16.4	Diagrama de secuencia _____	92
3.17	CU 17 Editar comentario de bitacora _____	93
3.17.1	Prototipo _____	93
3.17.2	Descripción _____	94
3.17.3	Diagrama de robustez _____	96
3.17.4	Diagrama de secuencia _____	97
3.18	CU 18 Eliminar comentario de bitácora _____	98
3.18.1	Prototipo _____	98
3.18.2	Descripción _____	99
3.18.3	Diagrama de robustez _____	100
3.18.4	Diagrama de secuencia _____	101
3.19	CU 19 Generar Pdf de la bitacora _____	102
3.19.1	Prototipo _____	102
3.19.2	Descripción _____	103
3.19.3	Diagrama de robustez _____	104
3.19.4	Diagrama de secuencia _____	104
3.20	CU 20 realizar cobro _____	105
3.20.1	Prototipo _____	105
3.20.2	Descripción _____	107
3.20.3	Diagrama de robustez _____	108

3.20.4	Diagrama de secuencia	109
3.21	CU 21 Consultar pedidos kiosko	110
3.21.1	Prototipo	110
3.21.2	Descripción	111
3.21.3	Diagrama de robustez	112
3.21.4	Diagrama de secuencia	113
3.22	CU 22 Realizar corte de caja	114
3.22.1	Prototipo	114
3.22.2	Descripción	115
3.22.3	Diagrama de robustez	116
3.22.4	Diagrama de secuencia	117
3.23	CU 23 Generar Pdf de movimientos diarios	118
3.23.1	Prototipo	118
3.23.2	Descripción	118
3.23.3	Diagrama de robustez	120
3.23.4	Diagrama de secuencia	121
3.24	CU 24 cancelar pedido	122
3.24.1	Prototipo	122
3.24.2	Descripción	123
3.24.3	Diagrama de robustez	124
3.24.4	Diagrama de secuencia	125
3.25	CU 25 consultar menús registrados	126
3.25.1	Prototipo	126
3.25.2	Descripción	127
3.25.3	Diagrama de robustez	128
3.25.4	Diagrama de secuencia	128
3.26	CU 26	129
3.26.1	Prototipo	129
3.26.2	Descripción	130
3.26.3	Diagrama de robustez	131
3.26.4	Diagrama de secuencia	132
3.27	Modelo de Dominio	133
3.28	Arquitectura Técnica	134
3.28.1	Modelo de paquetes	134
3.28.2	El modelo de despliegue	135
3.28.3	Modelo de datos	136
4	Pruebas	137
4.1	Pruebas de unidad	137
4.2	Pruebas de integración	146
4.3	Pruebas de sistema	157

Conclusiones	163
Bibliografía	164
Apéndice A Diagrama de clases	165
Apéndice B Diccionario de Datos	166
Apéndice C Tareas realizadas	170
Apéndice D. Estándar de codificación	171
Ilustración 1 Diagrama de casos de uso	19
Ilustración 2 Actividades designadas a primera entrega (40%)	21
Ilustración 3 Actividades designadas para la primera entrega (40%)	22
Ilustración 4 Actividades designadas para la primera entrega (40%)	22
Ilustración 5 Actividades designadas para la segunda entrega (90%).....	23
Ilustración 6 Actividades designadas para la segunda entrega (90%).....	23
Ilustración 7 Actividades designadas para la segunda entrega (90%).....	24
Ilustración 8. Actividades designadas tercera entrega (100%).....	24
Ilustración 9 Prototipo CU-01.....	25
Ilustración 10 Diagrama de robustez CU-01.....	27
Ilustración 11 Diagrama de secuencia CU-01.....	28
Ilustración 12 Prototipo CU-02.....	29
Ilustración 13 Diagrama de robustez CU-02.....	31
Ilustración 14 Diagrama de secuencia CU-02	32
Ilustración 15 Prototipo CU-03	33
Ilustración 16 Diagrama de robustez CU-03	36
Ilustración 17 Diagrama de secuencia CU-03	37
Ilustración 18 Prototipo CU-04	38
Ilustración 19 Diagrama de robustez CU-04	39
Ilustración 20 Diagrama de secuencia CU-04	40
Ilustración 21 Prototipo CU-05	40
Ilustración 22 Prototipo CU-05	41
Ilustración 23 Prototipo CU-05	41
Ilustración 24 Diagrama de robustez CU-05	43
Ilustración 25 Diagrama de secuencia CU-05	44
Ilustración 26 Prototipo CU-06	45
Ilustración 27 Prototipo CU-06	45
Ilustración 28 Diagrama de robustez CU-06	47
Ilustración 29 Diagrama de secuencia CU-06	48
Ilustración 30 Prototipo CU-07	49
Ilustración 31 Prototipo CU-07	49
Ilustración 32 Prototipo CU-07	50

Ilustración 33 Diagrama de robustez CU-07	52
Ilustración 34 Diagrama de secuencia CU-07	52
Ilustración 35 Prototipo CU-08	53
Ilustración 36 Diagrama de robustez CU-08	55
Ilustración 37 Diagrama de secuencia CU-08	56
Ilustración 38 Prototipo CU-09	57
Ilustración 39 Prototipo CU-09	57
Ilustración 40 Prototipo CU-09	58
Ilustración 41 Diagrama de robustez CU-09	60
Ilustración 42 Diagrama de secuencia CU-09	61
Ilustración 43 Prototipo CU-10	62
Ilustración 44 Diagrama de robustez CU-10	64
Ilustración 45 Diagrama de secuencia CU-10	65
Ilustración 46 Prototipo CU-11	66
Ilustración 47 Diagrama de robustez CU-11	68
Ilustración 48 Diagrama de secuencia CU-11	68
Ilustración 49 Prototipos CU-12	69
Ilustración 50 Prototipos CU-12	69
Ilustración 51 Prototipos CU-12	70
Ilustración 52 Diagrama de robustez CU-12	72
Ilustración 53 Diagrama de secuencia CU-12	73
Ilustración 54 Prototipo CU-13	74
Ilustración 55 Prototipo CU-13	74
Ilustración 56 Prototipo CU-13	75
Ilustración 57 Diagrama de robustez CU-13	77
Ilustración 58 Diagrama de secuencia CU-13	78
Ilustración 59 Prototipo CU-14	79
Ilustración 60 Prototipo CU-14	79
Ilustración 61 Prototipo CU-14	80
Ilustración 62 Diagrama de robustez CU-14	82
Ilustración 63 Diagrama de secuencia CU-14	83
Ilustración 64 Prototipo CU-15	84
Ilustración 65 Prototipo CU-15	84
Ilustración 66 Diagrama robustez CU-15	86
Ilustración 67 Diagrama secuencia CU-15	87
Ilustración 68 Prototipo CU-16	88
Ilustración 69 Prototipo CU-16	89
Ilustración 70 Prototipo CU-16	89
Ilustración 71 Diagrama de robustez CU-16	91
Ilustración 72 Diagrama de secuencia CU-16	92
Ilustración 73 Prototipo CU-17	93
Ilustración 74 Prototipo CU-17	94

Ilustración 75 Diagrama robustez CU-17	96
Ilustración 76 Diagrama secuencia CU-17	97
Ilustración 77 Prototipo CU-18	98
Ilustración 78 Prototipo CU-18	99
Ilustración 79 Diagrama de robustez CU-18	100
Ilustración 80 Diagrama de CU-18	101
Ilustración 81 Prototipo CU-19	102
Ilustración 82 Diagrama de robustez CU-19	104
Ilustración 83 Diagrama de secuencia CU-19	104
Ilustración 84 Prototipo CU-20	105
Ilustración 85 Prototipo CU-20	105
Ilustración 86 Prototipo Ticket	106
Ilustración 87 Diagrama Robustez CU-20	108
Ilustración 88 Diagrama secuencia CU-20	109
Ilustración 89 Prototipo CU-21	110
Ilustración 90 Diagrama de robustez CU-21	112
Ilustración 91 Diagrama secuencia CU-21	113
Ilustración 92 Prototipo CU-22	114
Ilustración 93 Diagrama robustez CU-22	116
Ilustración 94 Diagrama secuencia CU-22	117
Ilustración 95 Prototipo CU-23	118
Ilustración 96 Diagrama robustez CU-23	120
Ilustración 97 Diagrama secuencia CU-23	121
Ilustración 98 Prototipo CU-24	122
Ilustración 99 Prototipo CU-24	122
Ilustración 100 Diagrama de robustez CU-24	124
Ilustración 101 Diagrama de secuencia CU-24	125
Ilustración 102 Prototipo CU-25	126
Ilustración 103 Prototipo CU-25	126
Ilustración 104 Prototipo CU-25	127
Ilustración 105 Diagrama de robustez CU-25	128
Ilustración 106 Diagrama de secuencia CU-25	129
Ilustración 107 Prototipo CU-26	129
Ilustración 108 Prototipo CU-26	130
Ilustración 109 Diagrama robustez CU-26	131
Ilustración 110 Diagrama secuencia CU-26	132
Ilustración 111 Diagrama del dominio	133
Ilustración 112 Diagrama de paquetes	134
Ilustración 113 Diagrama de despliegue	135
Ilustración 114 Modelo de datos	136
Ilustración 115 Pruebas Modulo FoodController - Client	137
Ilustración 116 Pruebas Modulo MenuController – Client	138

Ilustración 117 Pruebas Modulo CashCutController - Employee	138
Ilustración 118 Pruebas Modulo DishOrderController - Employee.....	139
Ilustración 119 Pruebas Modulo EmployeeController - Employee	140
Ilustración 120 Pruebas Modulo FoodController - Employee	141
Ilustración 121 Pruebas Modulo FoodOrderController - Employee.....	143
Ilustración 122 Pruebas Modulo FoodOrderController - Employee.....	143
Ilustración 123 Pruebas IngredientController - Employee	144
Ilustración 124 Pruebas Modulo LogbookController - Employee	144
Ilustración 125 Pruebas Modulo SetMenuController - Employee	145
Ilustración 126 Pruebas totales	145
Ilustración 127 CP.01 Empleado ingresa al sistema a hacer un pedido	148
Ilustración 128 CP.01 Empleado realiza el cobro de un pedido.....	149
Ilustración 129 CP.01 Empleado consulta pedidos	149
Ilustración 130 CP.01 Empleado modifica pedido a "Hecho"	150
Ilustración 131 CP.01 Empleado modifica pedido a "Entregado".....	150
Ilustración 132 CP.02 Empleado registra ingrediente.....	151
Ilustración 133 CP.02 Empleado consulta ingrediente registrado	151
Ilustración 134 CP.02 Empleado agrega comentario a bitácora	152
Ilustración 135 CP.02 Gerente consulta reporte de inventario	152
Ilustración 136 CP.02 Gerente descarga reporte de inventario.....	153
Ilustración 137 CP.02 Gerente consulta reporte de bitacora.....	153
Ilustración 138 CP.02 Gerente descarga reporte de bitácora	154
Ilustración 139 CP.03 Empleado consulta corte de caja	154
Ilustración 140 CP.03 Empleado realiza corte de caja	155
Ilustración 141 CP.03 Gerente consulta reporte de movimientos diarios	155
Ilustración 142 CP.03 Gerente descarga reporte de movimientos diarios	156
Ilustración 143 CP.03 Descarga exitosa del reporte	156
Ilustración 144 CP.01 Registrar cajero	159
Ilustración 145 CP.01 Iniciar sesión como cajero.....	159
Ilustración 146 CP.02 Registrar jefe de cocina	160
Ilustración 147 CP.02 Iniciar sesión como jefe de cocina	160
Ilustración 148 CP.03 Registrar personal de cocina.....	161
Ilustración 149 CP.03 Iniciar sesión como personal de cocina	161
Ilustración 150 CP.04 Iniciar sesión como gerente	162
Ilustración 151 Diagrama de clases	165
 Tabla 1. Descripción CU-01.....	26
Tabla 2. Descripción CU-02.....	31
Tabla 3 Descripción CU-03.....	35
Tabla 4. Descripción CU-04.....	39
Tabla 5 Descripción CU-05.....	43

Tabla 6 Descripción CU-06.....	47
Tabla 7. Descripción CU-07.....	51
Tabla 8. Descripción CU-08.....	54
Tabla 9 Descripción CU-09.....	59
Tabla 10 Descripción CU-10.....	63
Tabla 11 Descripción CU-11.....	67
Tabla 12Descripción CU-12.....	72
Tabla 13 Descripción CU-13.....	77
Tabla 14 14 Descripción CU-14.....	81
Tabla 1515 Descripción CU-15.....	85
Tabla 16. Descripción CU-16.....	90
Tabla 17. Descripción CU-17.....	95
Tabla 18. Descripción CU-18.....	100
Tabla 19. Descripción CU-19.....	103
Tabla 20 Descripción CU-20.....	107
Tabla 21 Descripción CU-21.....	111
Tabla 22. Descripción CU-22.....	115
Tabla 23. Descripción CU-23.....	119
Tabla 24. Descripción CU-24.....	123
Tabla 25. Descripción CU-25.....	128
Tabla 26. Descripción CU-26.....	131
Tabla 27. Pruebas de integración	148
Tabla 28. Pruebas de sistema	158
Tabla 29. Diccionario de datos - CashCut	166
Tabla 30. Diccionario de datos - Employee	166
Tabla 31. Diccionario de datos - Employee_FoodOrder	166
Tabla 32. Diccionario de datos - FoodOrder	167
Tabla 33. Diccionario de datos - FoodOrder_MenuCard.....	167
Tabla 34. Diccionario de datos - FoodOrder_SetMenu	167
Tabla 35. Diccionario de datos - Ingredient.....	167
Tabla 36. Diccionario de datos - Logbook.....	168
Tabla 37. Diccionario de datos - MenuCard	168
Tabla 38. Diccionario de datos - SetMenu.....	168
Tabla 39. Diccionario de datos - User.....	169

Introducción

Antecedentes

El comedor recientemente inaugurado ofrece desayunos y comidas a precios accesibles para los estudiantes, la comunidad universitaria y el público. El comedor cuenta con un sistema básico para la venta de insumos y el corte de caja, pero carece de funcionalidades avanzadas para la gestión detallada de inventarios y reservas. Los ingredientes perecederos se reciben a diario y los no perecederos semanalmente, y cualquier sobrante de comida se maneja siguiendo protocolos de refrigeración y registro de mermas. Las reservas de comidas solo se permiten el mismo día, lo que limita la planificación a largo plazo tanto para los clientes como para la administración del comedor.

Contexto y Situación Actual

El sistema para el comedor universitario Komalli surge dentro del contexto de la Experiencia Educativa Desarrollo de Software con la finalidad de brindar una mejora al actual sistema del comedor. El propósito principal era realizar un software que permitiera a los empleados realizar sus actividades de seguimiento rutinario para el buen funcionamiento del comedor, pero, durante lalicitación, se descubrió que ya tenía un sistema, por lo que el propósito cambiaría dicho software.

Para complementar el sistema se busca implementar funcionalidad para la gestión de inventarios, que permita un seguimiento detallado de los insumos recibidos y utilizados diariamente, incorporar el registro de comentarios en bitácora para poder realizar el registro sobre mermas, algunos incidentes u otras situaciones que se presenten en el día a día del funcionamiento del comedor Komalli, además de ofrecer reportes que incorporen la información realizada en cuanto al registro en inventario, comentarios registrados en bitácora y registro de movimientos diarios dentro del sistema.

Para facilitar el flujo de actividades del personal de comedor, se busca implementar un Kiosko, con la finalidad de que los clientes que requieran algún alimento puedan realizar su pedido desde el allí sin que tengan que ir a caja a solicitarlo, con esta alternativa se busca movilizar el procedimiento de realizar un pedido ya que solo se recogerían y pagarían los alimentos.

Junto a esto, se busca un diseño de una interfaz intuitiva, fácil de usar y atractiva para el uso de los empleados y clientes, mejorando la rapidez y precisión en la toma de pedidos y la gestión diaria.

Justificación

La finalidad del desarrollo del sistema de comedor universitario Komalli es realizar una mejora del actual sistema del comedor, buscando optimizar diversas áreas clave del comedor para beneficiar a todos quienes lo utilicen.

El personal comedor realizará un seguimiento detallado de los insumos recibidos y utilizados diariamente, permitiendo reducir errores y asegurando la disponibilidad de los ingredientes. La implementación de una bitácora busca mejorar la calidad del servicio buscando el registro de

mermas, incidentes y otras situaciones. En cuanto a los reportes generados dentro del sistema, permitirá que el gerente pueda mantener una planificación y una toma de decisiones mejorada de acuerdo con lo que se obtiene dentro del histórico del sistema.

El personal UV y estudiantes tendrán el beneficio que al pedir en Kiosko se puedan realizar sus pedidos de una forma más rápida, reduciendo los tiempos de espera en cajero, además de que al presentar una interfaz intuitiva y atractiva a la vista se tenga una experiencia más agradable y accesible.

Objetivos

Objetivos generales

Dentro del objetivo general se encuentra la mejora de la gestión del comedor universitario Komalli para optimizar el seguimiento de insumos, la gestión de reservas y sobre todo priorizando la experiencia de usuario, brindando a empleados, maestros y estudiantes.

Objetivos específicos

- Seguimiento de inventario - Desarrollar una funcionalidad que permita el seguimiento de los insumos recibidos y utilizados diariamente.
- Mermas e incidentes - implementación de un módulo que permita el registro de comentarios en bitácora que documente mermas, incidentes y otras situaciones diarias.
- Reportes - Generar reportes dentro del sistema que detallen movimientos diarios dentro del sistema, como el registro de la llegada de ingredientes a inventario, las incidencias ocurridas en las actividades de comedor, registro de mermas, entre otros.
- Pedidos a través de Kiosko - permitir a los clientes realizar sus órdenes de forma autónoma, reduciendo los tiempos de espera y descongestionando la caja.
- Experiencia de usuario - diseñar una interfaz atractiva, accesible tanto para empleados como para clientes, facilitando la toma de pedidos y gestión diaria.

Métodología

Para el desarrollo del sistema Komalli, se empleará el método Iconix propuesto por el plan de estudios de la experiencia educativa Desarrollo de Software, debido a su enfoque orientado a objetos que se destaca por su eficiencia. Este método abarca todas las etapas esenciales del desarrollo de software, desde la conceptualización hasta la implementación asegurando la calidad del producto final.

Sus etapas son:

1. Modelado del dominio:

- Análisis de requerimientos - se realizó un análisis detallado de los requerimientos del sistema mediante entrevista con el personal del comedor universitario, el contemplar las opiniones de los estudiantes, permitiendo entender las necesidades y expectativas tanto empleados como estudiantes y personal UV.
- Diagrama de casos de uso - la elaboración de diagramas de casos de uso son útiles para identificar y definir las interacciones entre usuarios y el sistema. Estos diagramas ayudan a poder visualizar las funcionalidades necesarias y a establecer una línea base para su desarrollo.

2. Diseño robusto:

- Modelado del dominio - se crearon diagramas tanto de dominio como de clases que representan las identidades principales del sistema y sus relaciones, este modelado es útil para estructurar el sistema y definir claramente las responsabilidades de cada componente.
- Diagramas de robustez - el diseño de diagramas de robustez para cada caso de uso empleando los objetos del dominio, controladores, almacenes y vistas fue esencial para asegurar el diseño adecuadamente entre las interacciones de los componentes.

3. Diseño detallado:

- Diagramas de secuencia - los diagramas de secuencia por cada caso de uso detallarán los flujos y mensajes entre objetos, garantizando que las interacciones sean correctas en su momento de implementarlas.
- Diseño de prototipos - enfocadas en la usabilidad y experiencia del usuario. Se busca que sean intuitivas y fácil de usar, logrando la satisfacción del usuario final.

4. Implementación y pruebas

- Codificación - gracias a la creación de diagramas y modelos creados, se procederá a la creación de código.
- Pruebas unitarias y de integración - la realización de pruebas unitarias se necesitarán para verificar que cada componente funcione de manera independiente. Mientras que las de integración para asegurar el correcto funcionamiento de los componentes de forma conjunta y cumpliendo con los requerimientos establecidos.

Estructura del Trabajo

El desglose de tareas detalla las actividades que se llevarán a cabo o se han llevado a cabo durante el desarrollo del proyecto de software de Komalli. En él se especifican las tareas, los recursos humanos asignados, así como el tiempo y el porcentaje de avance en cada actividad.

Se puede consultar las tareas realizadas por el equipo dentro del Apéndice C Tareas realizadas.

1 Requerimientos

Este apartado contendrá información necesaria para el desarrollo de nuestro proyecto de software, lo que incluye la definición de nuestro alcance, lo cual significa qué objetivos se plantean para lograr el buen desarrollo y funcionalidad final del sistema, para lograr esto, se han definido requisitos, de los cuales están contenidos en dicho apartado, que nos ayudarán a guiarnos para saber si se está cumpliendo con las necesidades identificadas y establecidas para cada uno de los usuarios y secciones localizadas.

1.1 Alcance del Proyecto

Es un alcance amplio pues se busca abarcar diversas áreas clave del funcionamiento del comedor universitario Komalli. Dentro del objetivo general es desarrollar una mejora del sistema de gestión del comedor Komalli, optimizando su administración de inventarios, los pedidos y mejorando la experiencia de usuario lo cual incluye personal de comedor, personal UV, estudiantes y público general.

Se busca realizar funciones de inventarios que permitirá el registro detallado de insumos recibidos y utilizados día a día, también se busca implementar el registro de incidentes, mermas y otras situaciones, facilitando la generación de reportes para apoyar la toma de decisiones. El implementar una función para realizar pedidos de forma autónoma para el cliente con el objetivo de descongestionar la realización de pedidos en caja.

El enfoque en el diseño de interfaz de usuario intuitiva y atractiva tanto para el personal de comedor, personal UV y estudiantes ayudará a la precisión de pedidos y su gestión de forma diaria, así mismo con la implementación de reportes se permite el acceso a datos históricos y mejorando la planificación y toma de decisiones del gerente del comedor.

Komalli busca no solo optimizar el funcionamiento del comedor universitario, sino también mejorar significativamente la experiencia de todos sus usuarios, proporcionando un sistema fácil de usar.

1.2 Clases y Características de usuarios

Cajero

El cajero es responsable de interactuar directamente con los clientes en el punto de venta, gestionando los pedidos y las transacciones financieras.

- El cajero puede registrar, modificar, eliminar y revisar comentarios en la bitácora para llevar un seguimiento de incidencias o eventos relevantes durante su turno.
- Gestiona las transacciones de pago de los clientes
- Toma y registra los pedidos realizados por los clientes en el sistema.

- Verifica y consulta los pedidos que han sido realizados a través de kioscos de autoservicio.
- Al final de su turno, realiza el corte de caja para cuadrar las ventas del día con el dinero en efectivo.

Personal cocina

Se encarga de preparar los alimentos y gestionar los menús, asegurando que los pedidos se realicen correctamente y a tiempo.

- Registra y gestiona comentarios sobre incidencias o eventos en la cocina.
- Administra los menús, incluyendo la creación, modificación y eliminación de elementos del menú.
- Actualiza el estado de los pedidos (por ejemplo, pendientes, hechos y entregados).

Jefe cocina

Supervisa al personal de cocina, gestiona el inventario de ingredientes y asegura que la operación culinaria sea eficiente.

- Gestiona comentarios sobre el funcionamiento de la cocina.
- Administra el inventario de ingredientes, incluyendo el registro, la actualización y la búsqueda de ingredientes para asegurar disponibilidad.

Gerente

Responsable de la administración general del establecimiento, incluyendo la gestión de usuarios, la generación de reportes y la supervisión de operaciones diarias.

- Administra las cuentas de usuario del personal, incluyendo la creación, modificación y consulta de información de usuarios.
- Crea reportes detallados sobre las actividades registradas en la bitácora, el estado del inventario y los movimientos financieros y operativos diarios.

Cliente

Interactúa con el sistema para realizar pedidos.

1.3 Requerimientos Funcionales

Requisitos funcionales divididos por módulos:

Usuarios

RF-01: El sistema Komalli debe permitir registrar a los empleados del comedor, se debe solicitar un nombre de usuario, rol dentro del comedor, numero de personal, así como un correo electrónico.

RF-02: Komalli debe permitir la consulta, así como la modificación de información registrada de los usuarios.

RF-03: Se debe permitir la búsqueda de usuarios por nombre, lo cual debe arrojar un resultado de búsqueda ordenado.

RF-04: Es necesario que se considere con mecanismos de autentificación segura donde se solicite el nombre de usuario y contraseña para poder acceder dentro del sistema.

Ingredientes

RF-05: Se debe tener la posibilidad de registrar nuevos ingredientes detallando su nombre, código, descripción, precio y cantidad disponible.

RF-06: Debe permitir la edición de la información de ingredientes registrados, excepto el código.

RF-07: Komalli debe permitir a los empleados del comedor universitario buscar ingredientes por nombre o código, y mostrar los resultados de forma ordenada.

RF-08: Se deben generar informes de inventario en formato PDF.

RF-09: Debe haber una función para comparar la cantidad física de productos con la cantidad registrada en el sistema ayudando así a detectar posibles discrepancias.

Pedidos

RF-10: El sistema debe permitir a los clientes realizar pedidos especificando productos y cantidades deseadas, así como el usuario solicitante y el estado del pedido.

RF-11: El sistema debe permitir a los usuarios editar pedidos existentes, agregar o eliminar productos y modificar cantidades.

RF-12: Se debe proporcionar una función para actualizar el estado de los pedidos, cambiándolos de pagado a entregado.

RF-13: El sistema debe permitir consultar los pedidos realizados en el kiosko y proceder a su cobro.

RF-14: El sistema debe permitir a los empleados del comedor cuyo rol sea cajero levantar pedidos

Cocina

RF-15: Debe permitir a los empleados visualizar la lista completa de pedidos a preparar, incluyendo detalles como fecha del pedido, nombre del usuario, productos solicitados y cantidad requerida.

RF-16: Se debe permitir a los empleados planificar y organizar los menús semanales.

RF-17: Komalli permite administrar por medio de una bitácora aquellos sucesos o imprevistos que puedan suscitarse a lo largo de la jornada laboral dentro del comedor.

Finanzas

RF-18: Se debe considerar el registro de ingresos diarios indicando la fecha y el monto recibido.

RF-19: Komalli debe permitir la realización de balances diarios para analizar los ingresos y gastos del día.

1.4 Requerimientos No Funcionales

Requisitos No funcionales

Seguridad:

RNF-01: El sistema deberá asegurar la privacidad y confidencialidad de los datos de los usuarios, como datos personales mediante la implementación de medidas de encriptación.

RNF-02: El sistema debe implementar un modelo de control de acceso basado en roles que restrinja el acceso a funciones y datos según el rol del usuario.

Usabilidad:

RNF-03: El sistema deberá contar con una interfaz de usuario intuitiva y fácil de usar iconografía conocida, donde un usuario logre completar una tarea en cinco clics.

Rendimiento:

RNF-04: El sistema deberá de ser capaz de manejar múltiples usuarios concurrentes durante horas de alta demanda (11:00 a.m. a 3:00 p.m.) soportando 30 usuarios simultáneos sin degradación del rendimiento.

RNF-05: El tiempo de respuesta para las acciones del sistema, como la carga de menús o la gestión de pedidos, deberá ser rápido y no exceder los 5 segundos en condiciones normales de carga.

Mantenibilidad

RNF-06: El sistema deberá estar separado de manera modular para lograr de esta forma una alta cohesión en cada módulo y un bajo acoplamiento entre estos, con la finalidad de que sea posible realizar modificaciones al código de manera sencilla.

RNF-07: En el sistema se deberá aplicar estándares de codificación uniformes y buenas prácticas de programación para mantener un código limpio, legible y más fácil de mantener a lo largo del tiempo.

1.5 Modelo de Casos de Uso

En la Ilustración 1 se muestra el cómo el usuario interactúa con el sistema. Esta clasificado por actores y en el lado izquierdo se ocupa un actor Personal del comedor que tiene relacionado a otros actores y casos de uso. Los actores que heredan de este de igual manera podrán utilizar los casos de uso que este tiene relacionado. De igual manera existe otro actor llamado Cliente donde el solo interactúa con un Kiosko y podrá seleccionar y generar sus propios pedidos. Por último, hay un actor Gerente que podrá administrar las operaciones que se realicen dentro del sistema y el de dar de alta a nuevos usuarios empleados.

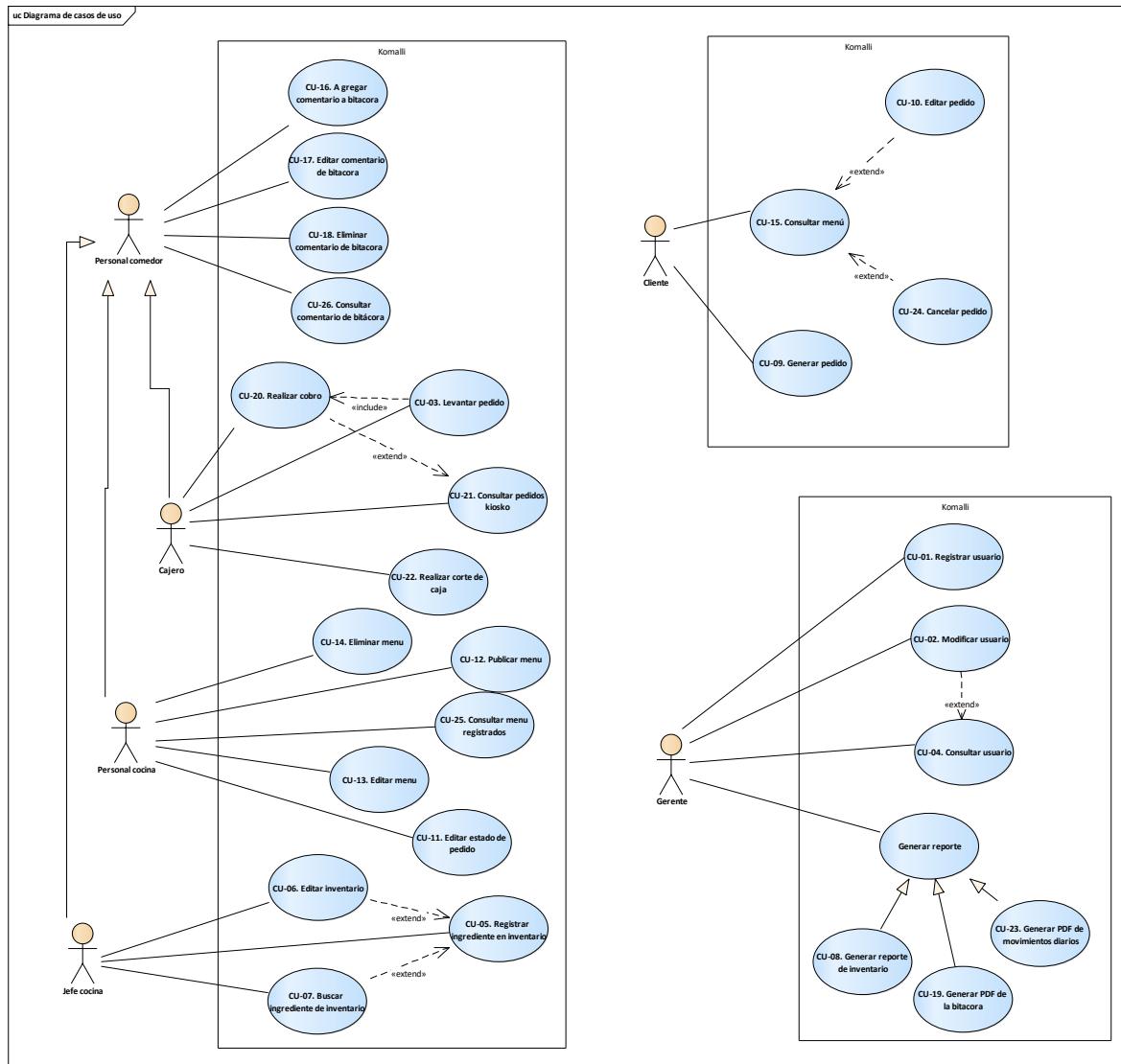


Ilustración 1 Diagrama de casos de uso

2 Administración del Proyecto

La administración del proyecto es esencial para asegurar el éxito y la organización efectiva de todas las actividades relacionadas con el desarrollo del sistema. Esta sección proporciona una visión integral de los elementos clave necesarios para gestionar y coordinar el proyecto de manera eficiente.

2.1 Lista de Casos de Uso

- CU01.- Registrar Usuario.
- CU02.- Modificar Usuario.
- CU03.- Levantar pedido
- CU04.- Consultar Usuario.
- CU05.- Registrar Ingrediente en inventario.
- CU06.- Editar Inventario.
- CU07.- Buscar Ingrediente en inventario.
- CU08.- Generar reporte de inventario.
- CU09.- Generar Pedido.
- CU10.-Editar Pedido.
- CU11.-Editar Estado del pedido.
- CU12.-Publicar menú.
- CU13.- Editar menú.
- CU14.- Eliminar menú.
- CU15.- Consultar menú.
- CU16.-Agregar comentario a bitácora
- CU17.-Editar comentario de bitácora
- CU18.- Eliminar comentario de bitácora.
- CU19.- Generar pdf de la bitácora.
- CU20.- Realizar cobro.
- CU21.- Consultar pedidos kiosko
- CU22.- Realizar corte de caja.
- CU23.- Generar PDF de movimientos diarios.
- CU24.-Cancelar pedido.
- CU25.- Consultar menú registrados.
- CU26.-Consultar comentario de bitácora

2.2 Lista de tareas

Para este proyecto se definieron tres revisiones, las cuales constan de definir las tareas que se realizaran para cumplir con el porcentaje de avance, la primera equivale al 40% del proyecto, la segunda al 90% y la tercera al 100%.

Para la primera revisión se definieron las siguientes tareas:

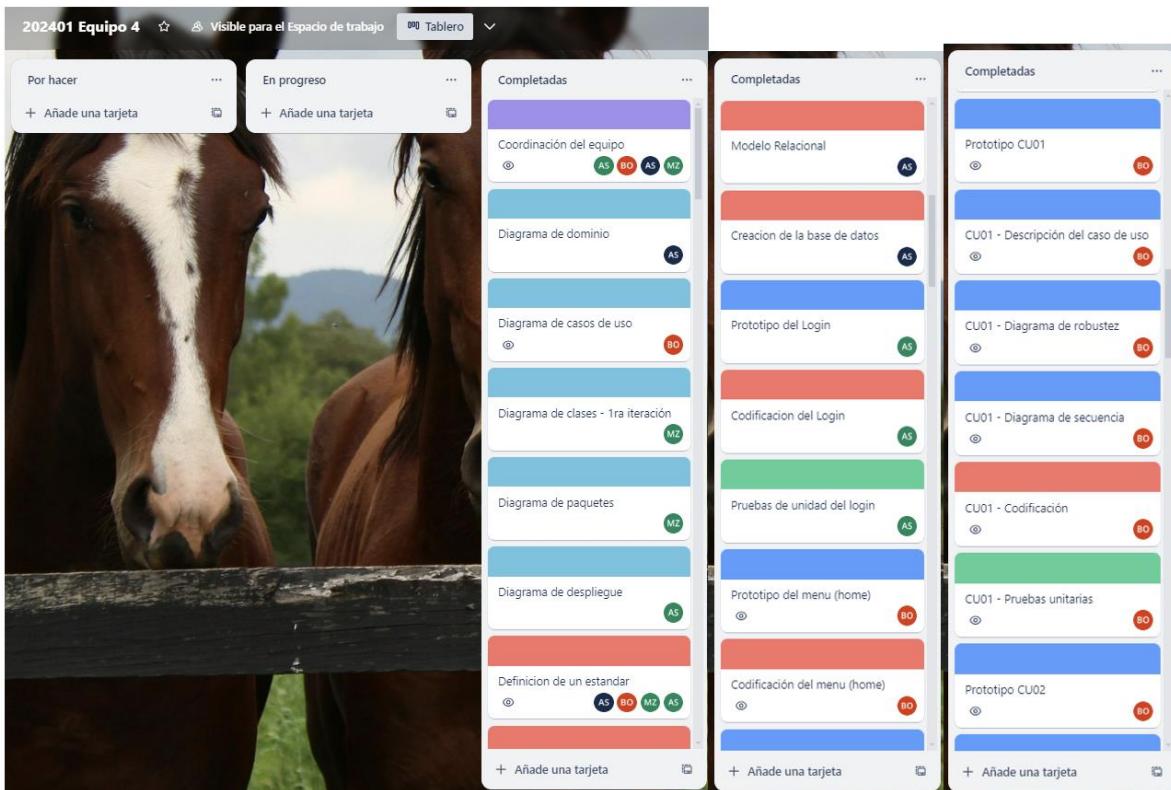
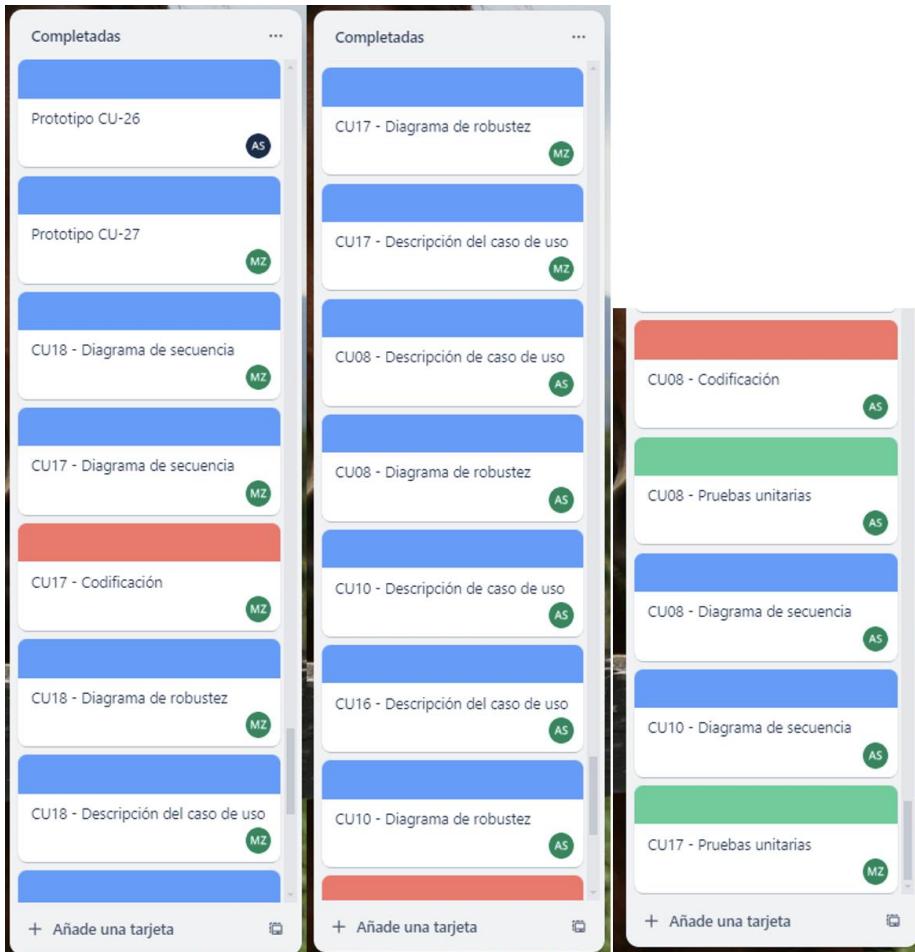
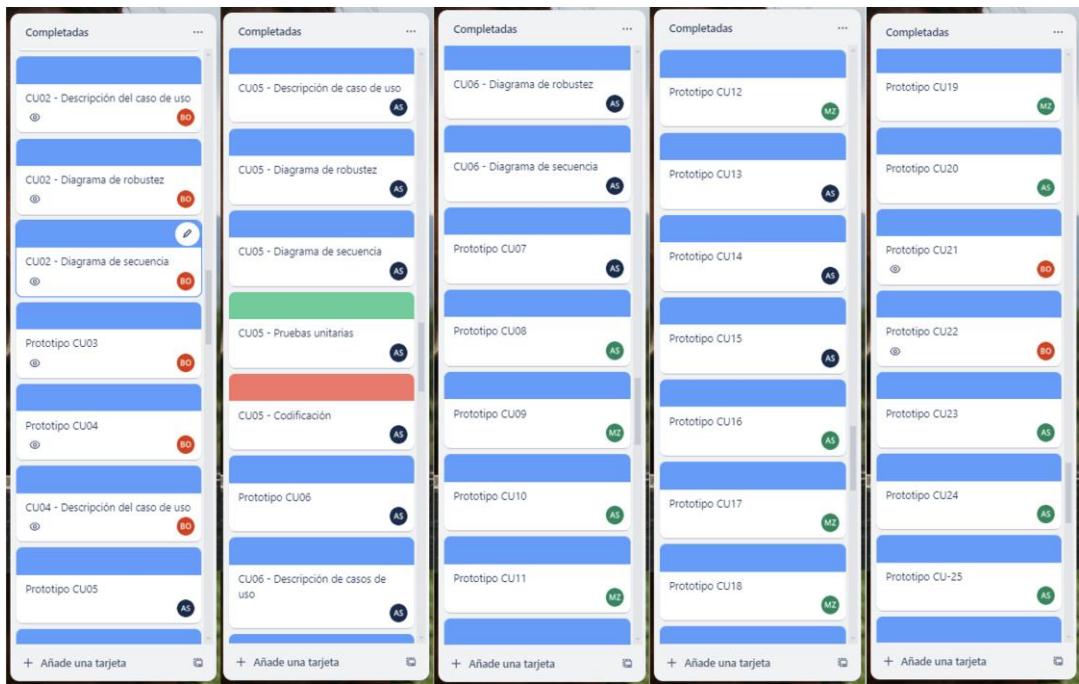


Ilustración 2 Actividades designadas a primera entrega (40%)



Para la segunda revisión se definieron las siguientes tareas:

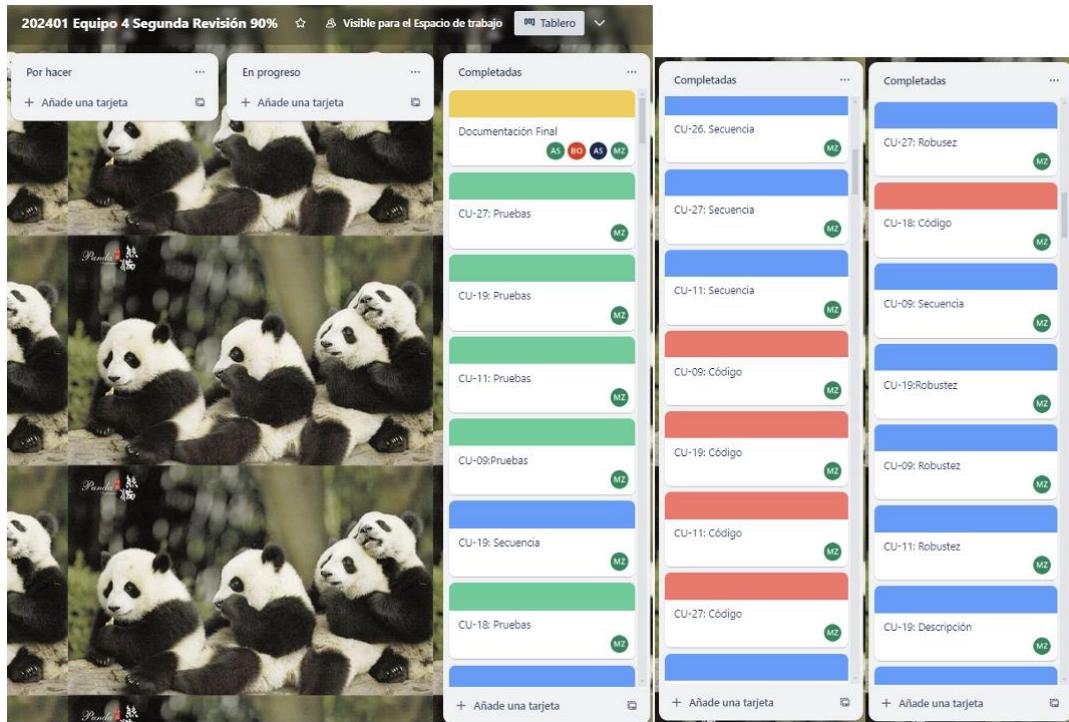


Ilustración 5 Actividades designadas para la segunda entrega (90%)

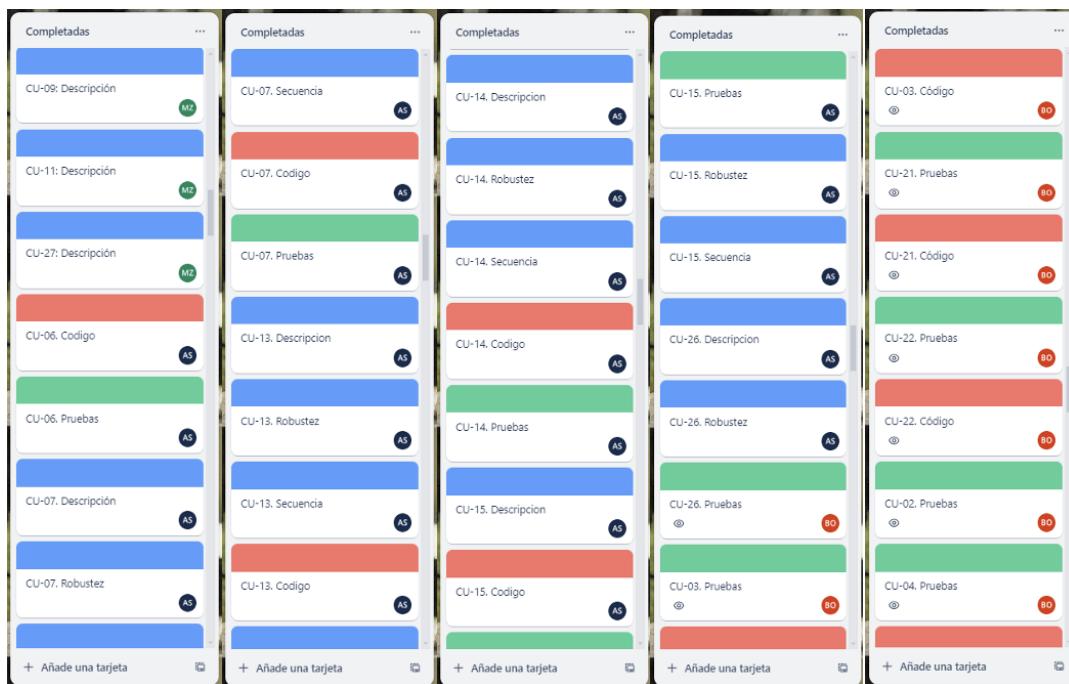


Ilustración 6 Actividades designadas para la segunda entrega (90%)

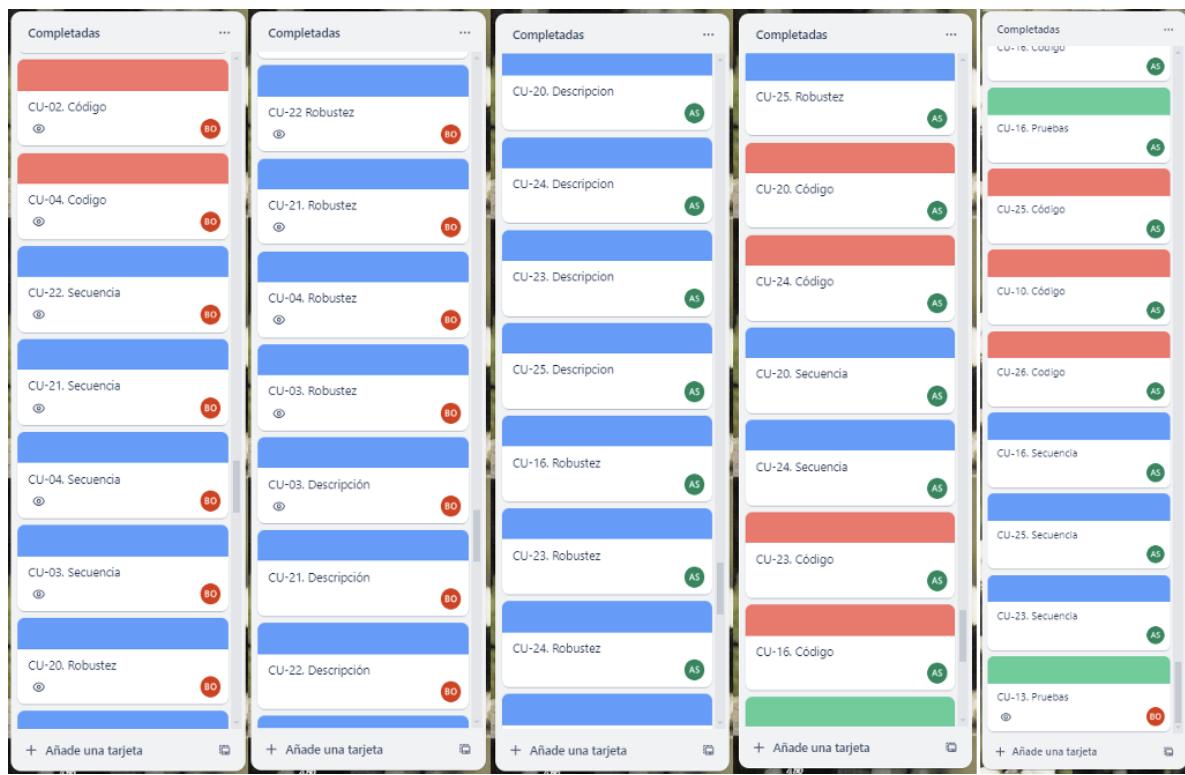


Ilustración 7 Actividades designadas para la segunda entrega (90%)

Por último, para la tercera entrega se definieron las siguientes tareas:

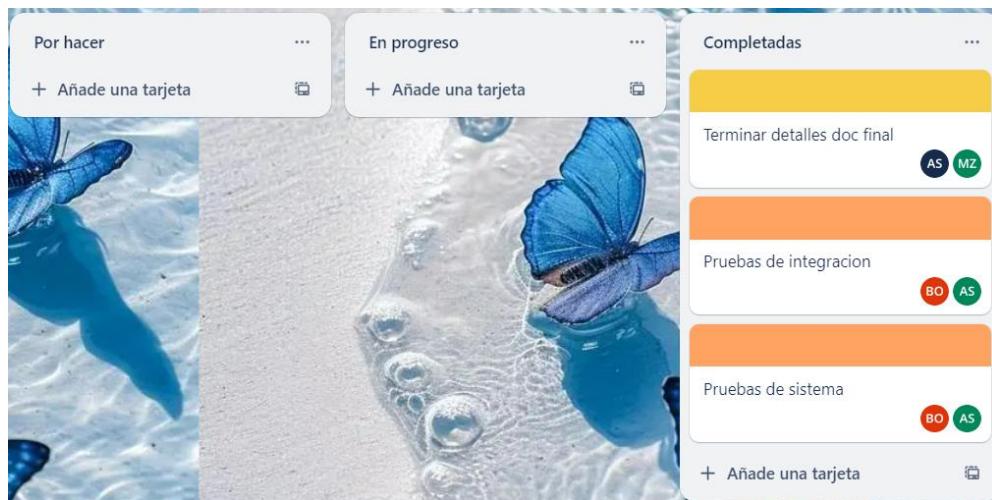


Ilustración 8. Actividades designadas tercera entrega (100%)

3 Análisis y Diseño

En los siguientes puntos que harán referencia al análisis y diseño de Komalli, se enfocará en detallar cada uno de los 26 casos de uso identificados, proporcionando los aspectos funcionales y su relación con los requisitos funcionales previamente descritos. En este apartado, se presentarán prototipos de las interfaces de usuario, descripciones detalladas de cada caso de uso, así como diagramas de robustez y secuencia. Los prototipos permitirán visualizar la interacción del usuario con el sistema, mientras que las descripciones ofrecerán una comprensión clara de los objetivos y flujos de cada caso de uso. Los diagramas de robustez mostrarán el comportamiento que tiene cada caso de uso entre la relación cliente-sistema, y los diagramas de secuencia ilustrarán la dinámica de las interacciones entre los diferentes componentes del sistema durante la ejecución de cada caso de uso. Este enfoque asegurará que todos los aspectos del diseño sean considerados, proporcionando una base para el desarrollo y la implementación del sistema.

3.1 CU 01 Registrar Usuario

3.1.1 Prototipo

El prototipo muestra la interfaz de usuario para el caso de uso CU-01. La barra lateral izquierda tiene un logo de Komalli y menús para 'Usuarios' (Consultar, Registrar) y 'Reportes'. Una barra superior indica que el usuario es Miriam Ramírez Zárate. El formulario principal titulado 'Registro de usuarios' contiene campos para Nombre, No. Personal, Correo y Rol, cada uno con un cuadro de texto correspondiente. Un botón 'Registrar' está al final del formulario.

Ilustración 9 Prototipo CU-01

3.1.2 Descripción

ID:	CU- 01
Nombre del CU:	Registrar usuario
Requisito funcional:	RF-01: El sistema Komalli debe permitir registrar a los empleados del comedor, se debe solicitar un nombre de usuario, rol dentro del comedor, numero de personal, asi como un correo electrónico.
Descripción:	Se registra un usuario en la base de datos del sistema
Actor(es):	Gerente (Primario)
Disparador:	El Gerente da clic en “Registrar” de la sección Usuarios del menú del sistema
Precondiciones:	PRE-01: El usuario a registrar no se encuentra previamente registrado en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI muestra la ventana “RegisterUser” con un formulario. 2. El GERENTE ingresa <i>nombre</i>, <i>correo</i>, <i>numeroPersonal</i>, y selecciona un <i>rol</i> y selecciona la opción “Registrar” de “RegisterUser”. 3. KOMALLI valida los datos ingresados del usuario en el formulario. (ver FA-01, ver FA-02). 4. KOMALLI guarda los datos del usuario en la base de datos, mostrando la alerta <u>GUI-REGISTRO-USUARIO-EXITOSO</u>. (ver EX-01). 5. El GERENTE da clic en la opción “Aceptar” de la alerta <u>GUI-REGISTRO-USUARIO-EXITOSO</u> para cerrar la alerta. 6. KOMALLI cierra la alerta <u>GUI-REGISTRO-USUARIO-EXITOSO</u>. 7. Termina el caso de uso
Flujos Alternos:	<p>FA-01: Datos Inválidos:</p> <ol style="list-style-type: none"> 1. KOMALLI detecta campos inválidos y muestra la alerta <u>GUI-DATOS-INVALIDOS</u>. 2. El GERENTE da clic en la opción “Ok” de la alerta <u>GUI-DATOS-INVALIDOS</u>. 3. KOMALLI cierra alerta <u>GUI-DATOS-INVALIDOS</u>. 4. Regresar al paso 2 del flujo normal. <p>FA-02: Datos Duplicados:</p> <ol style="list-style-type: none"> 1. KOMALLI detecta información duplicada y muestra la alerta <u>GUI-DATOS-DUPPLICADOS</u>. 2. El GERENTE da clic en la opción “Ok” de la alerta <u>GUI-DATOS-DUPPLICADOS</u>. 3. KOMALLI cierra alerta <u>GUI-DATOS- DUPLICADOS</u>. 4. Regresar al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El GERENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01. Un nuevo usuario es registrado en la base de datos del sistema.
Incluye:	N/A
Extiende:	N/A

Tabla 1. Descripción CU-01

3.1.3 Diagrama de robustez

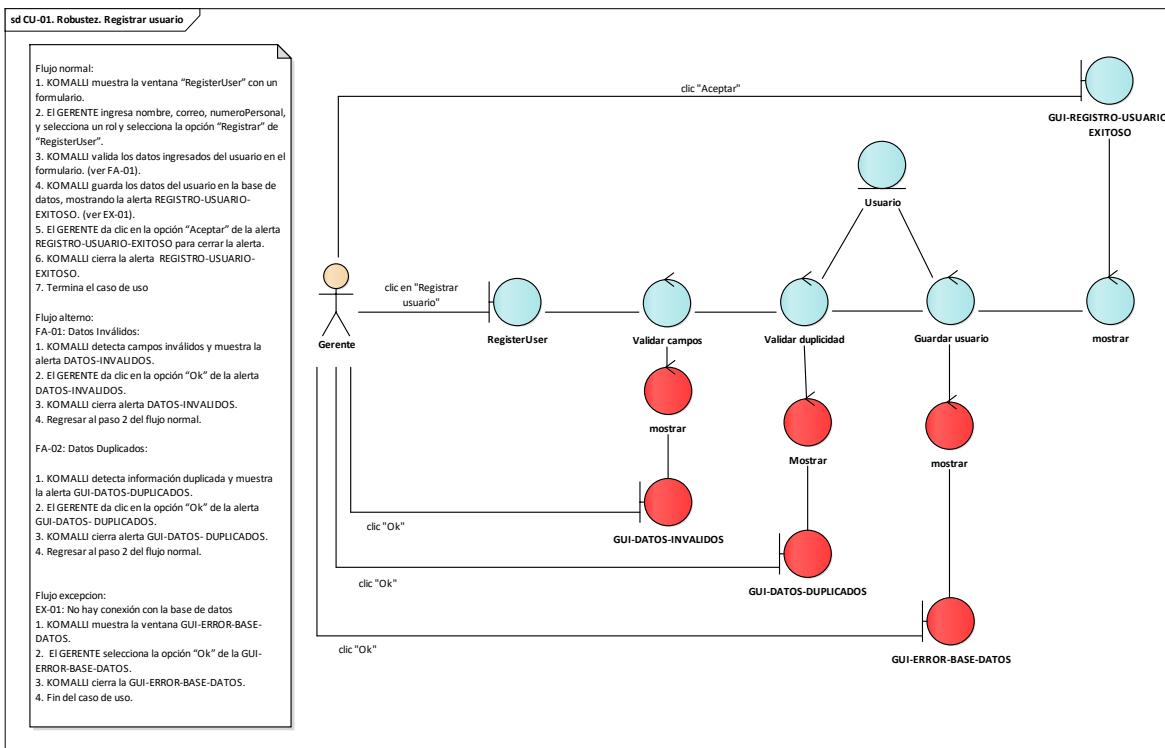


Ilustración 10 Diagrama de robustez CU-01

3.1.4 Diagrama de secuencia

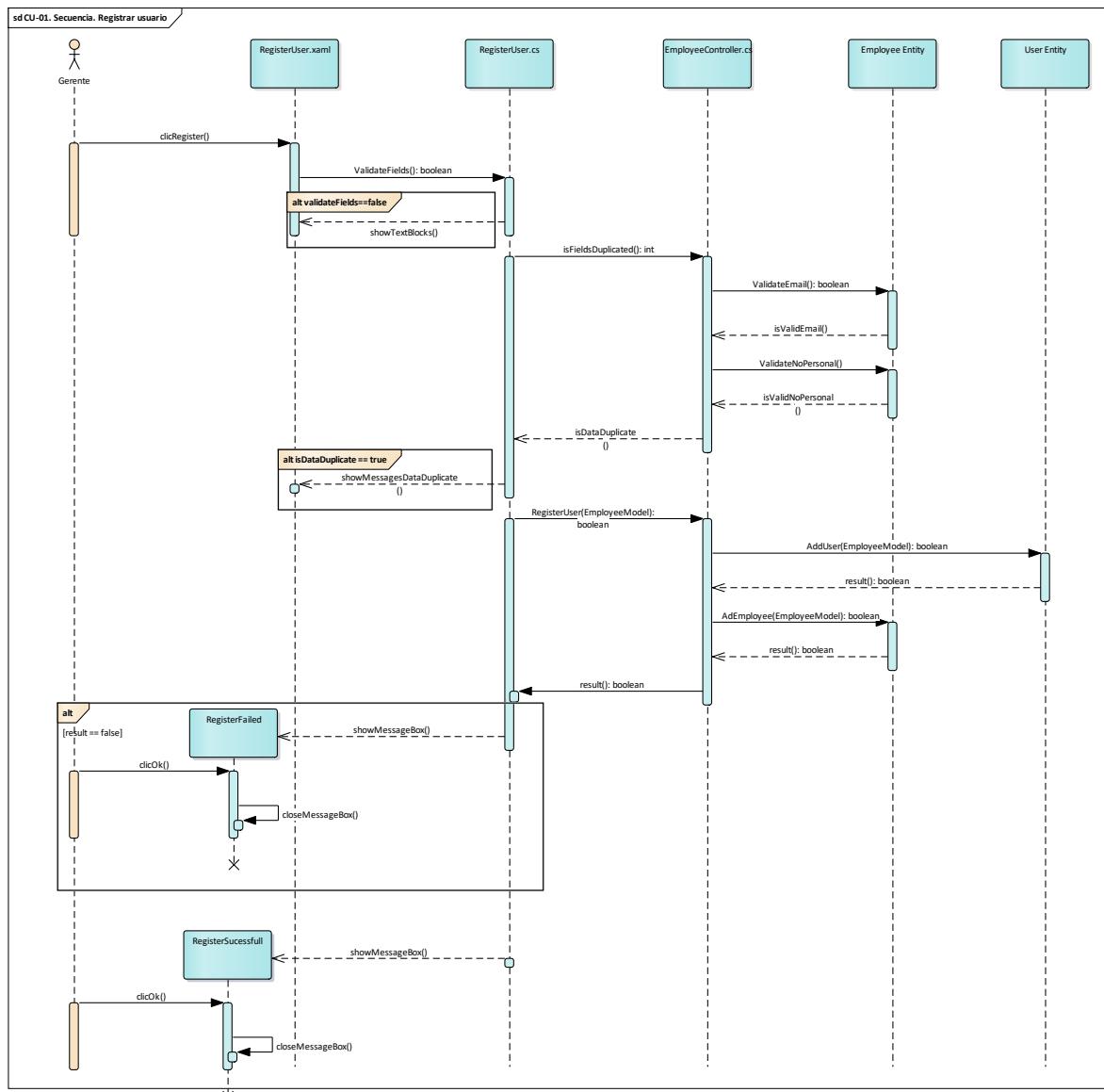


Ilustración 11 Diagrama de secuencia CU-01

3.2 CU 02 Modificar Usuario

3.2.1 Prototipo

El prototipo muestra una interfaz web con el logo "Komalli" en la parte superior izquierda. En la barra lateral izquierda, hay un menú con "Usuarios" (seleccionado), "Consultar" y "Registrar", y "Reportes". Una barra de herramientas en la parte inferior indica que el usuario actual es "Miriam Ramírez Zárate". El formulario principal se titula "Modificar usuario" y contiene los siguientes campos:

Nombre:	Miriam Ramírez Zárate
No. Personal:	484
Correo:	miri27.rz@gmail.com
Rol:	Cajero
Estatus:	Inactivo

Un botón "Modificar" se encuentra en la parte inferior del formulario.

Ilustración 12 Prototipo CU-02

3.2.2 Descripción

ID:	CU- 02
Nombre del CU:	Modificar usuario
Requisito funcional:	RF-02: Komalli debe permitir la consulta así como la modificación de información registrada de los usuarios.
Descripción:	Se modifican los datos de un usuario en la base de datos del sistema
Actor(es):	Gerente (Primario)
Disparador:	El Gerente da clic en “Modificar” en fila de un usuario en la tabla de la ventana “ConsultUsers”.
Precondiciones:	PRE-01: El usuario a modificar se encuentra previamente registrado en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI recupera de la base de datos los datos del usuario seleccionado y los muestra la ventana “ModifyUsers”. 2. El GERENTE modifica alguno de los datos y selecciona la opción “Modificar” de “ModifyUsers”. 3. KOMALLI valida los datos ingresados del usuario en el formulario. (ver FA-01, ver FA-02). 4. KOMALLI guarda los datos del usuario en la base de datos, mostrando la alerta <u>GUI-MODIFICAR-USUARIO-EXITOSO</u>. (ver EX-01). 5. El GERENTE da clic en la opción “Aceptar” de la alerta <u>GUI-MODIFICAR-USUARIO-EXITOSO</u> para cerrar la alerta. 6. KOMALLI cierra la alerta <u>GUI-MODIFICAR-USUARIO-EXITOSO</u>. 7. Termina el caso de uso
Flujos Alternos:	<p>FA-01: Datos Inválidos:</p> <ol style="list-style-type: none"> 1. KOMALLI detecta campos inválidos y muestra la alerta <u>GUI-DATOS-INVALIDOS</u>. 2. El GERENTE da clic en la opción “Ok” de la alerta <u>GUI-DATOS-INVALIDOS</u>. 3. KOMALLI cierra alerta <u>GUI-DATOS-INVALIDOS</u>. 4. Regresar al paso 2 del flujo normal. <p>FA-02: Datos Duplicados:</p> <ol style="list-style-type: none"> 1. KOMALLI detecta información duplicada y muestra la alerta <u>GUI-DATOS-DUPLICADOS</u>. 2. El GERENTE da clic en la opción “Ok” de la alerta <u>GUI-DATOS-DUPLICADOS</u>. 3. KOMALLI cierra alerta <u>GUI-DATOS- DUPLICADOS</u>. 4. Regresar al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El GERENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01. Los datos del usuario son modificados en la base de datos del sistema.

Incluye:	N/A
Extiende:	N/A

Tabla 2. Descripción CU-02

3.2.3 Diagrama de robustez

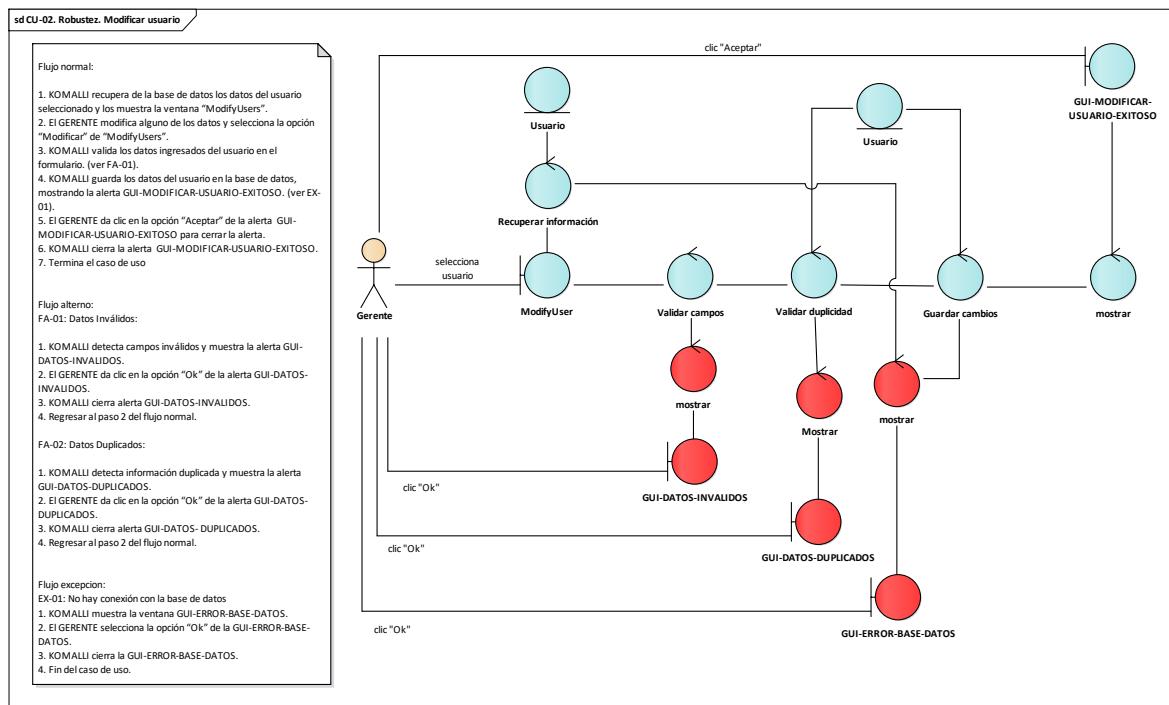


Ilustración 13 Diagrama de robustez CU-02

3.2.4 Diagrama de secuencia

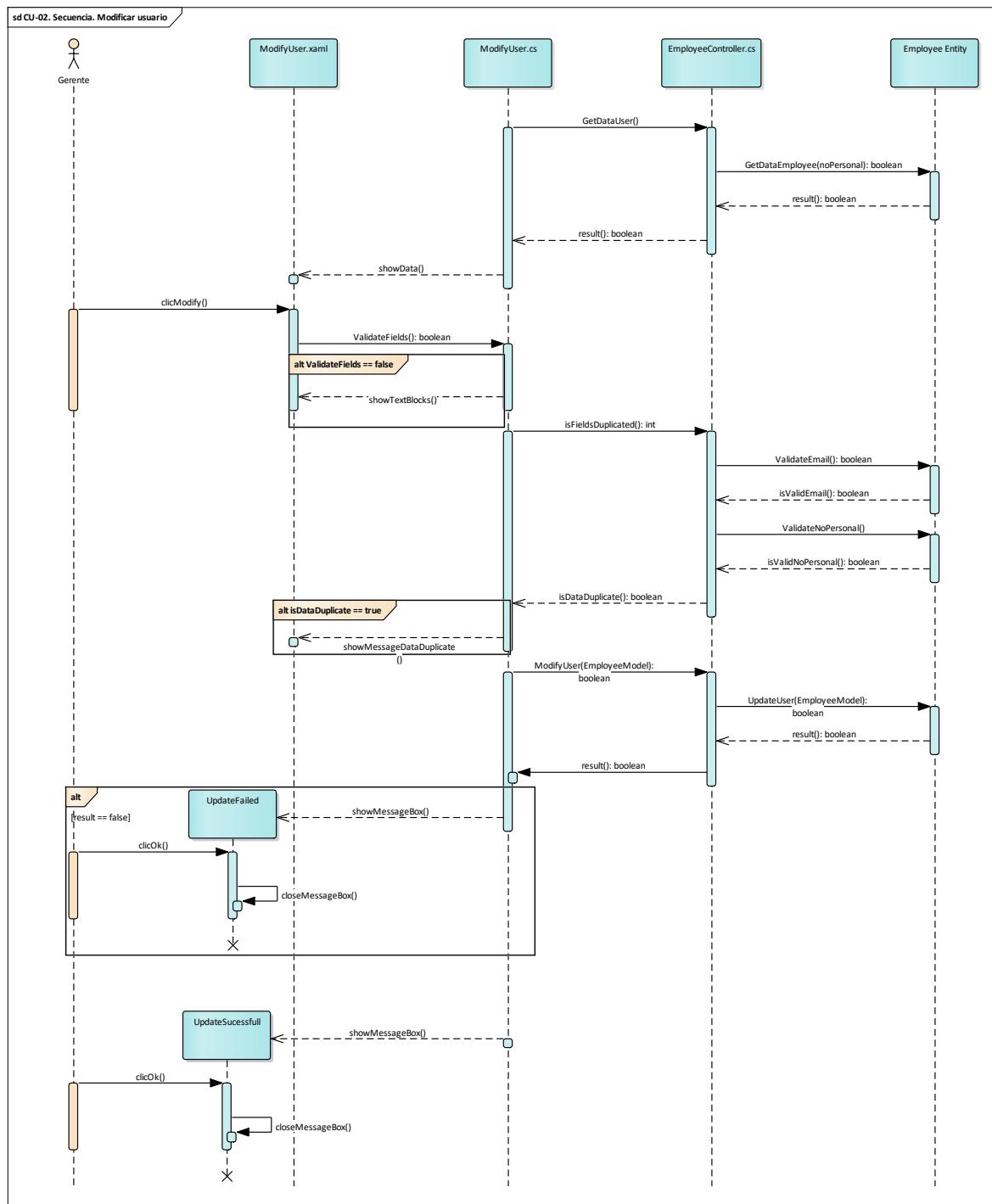


Ilustración 14 Diagrama de secuencia CU-02

3.3 CU 03 Levantar pedido

3.3.1 Prototipo



Ilustración 15 Prototipo CU-03

3.3.2 Descripción

ID:	CU-03
Nombre del CU:	Levantar pedido
Requisito funcional:	<p>RF-14: El sistema debe permitir a los empleados del comedor cuyo rol sea cajero levantar pedidos</p> <p>RF-11: El sistema debe permitir a los usuarios editar pedidos existentes, agregar o eliminar productos y modificar cantidades.</p>
Descripción:	Se desea seleccionar los platillos para realizar un pedido.
Actor(es):	Cajero (Primario)
Disparador:	El cajero da clic en la opción “Realizar pedido” de la sección Caja el menú principal.
Precondiciones:	PRE-01: Hay productos registrados en el sistema.
Flujo Normal	<ol style="list-style-type: none"> KOMALLI recupera de la base de datos los platillos que fueron previamente registrados y los separa en distintas categorías mostrando el nombre y precio del alimento y los muestra en la ventana “MakeOrder”. (ver EX-01) El CAJERO selecciona la opción “Agregar” de algún alimento de alguna de las categorías. (ver FA-01) (ver FA-02) (ver FA-04) KOMALLI agrega el alimento a la lista del pedido. (ver FA-01, FA-02). El CAJERO selecciona la opción “Cobrar”. KOMALLI registra en la base de datos del sistema el nuevo pedido mostrando la alerta <u>GUI-PEDIDO-REGISTRADO-EXITOSAMENTE</u>. (ver EX-01) El CAJERO da clic en la opción “Aceptar” de la alerta para <u>GUI-PEDIDO-REGISTRADO-EXITOSAMENTE</u> cerrarla. KOMALLI cierra la alerta <u>GUI-PEDIDO-REGISTRADO-EXITOSAMENTE</u>. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Aumentar la cantidad de un platillo</p> <ol style="list-style-type: none"> El CAJERO da clic en el ícono más de un platillo, de la lista del pedido. KOMALLI aumenta en uno la cantidad del platillo. (ver FA-03). El flujo continuo en el paso 4. <p>FA-02: Eliminar un platillo</p> <ol style="list-style-type: none"> El CAJERO da clic en el ícono de bote de basura de un platillo, de la lista del pedido. KOMALLI elimina de la lista el platillo seleccionado. El flujo continuo en el paso 4. <p>FA-03: Disminuir la cantidad de un platillo</p> <ol style="list-style-type: none"> El CAJERO da clic en el ícono de menos de un platillo, de la lista del pedido. KOMALLI disminuye en uno la cantidad del platillo. El flujo continuo en el paso 4. <p>FA-04: Borrar pedido</p> <ol style="list-style-type: none"> El CAJERO da clic en el ícono de borrar, de la lista del pedido. KOMALLI muestra una alerta <u>GUI-BORRAR-PEDIDO</u> con dos opciones, “Si” y “No”. El CAJERO selecciona la opción “Si” de la alerta <u>GUI-BORRAR-PEDIDO</u>. (ver FA-05). KOMALLI cierra la alerta <u>GUI-BORRAR-PEDIDO</u> y borra el registro de la lista de pedido. Regresa al paso 2 del flujo normal. <p>FA-05: Cancelar borrar pedido</p> <ol style="list-style-type: none"> El CAJERO selecciona la opción “No” de la alerta <u>GUI-BORRAR-PEDIDO</u>. (ver FA-05). KOMALLI cierra la alerta <u>GUI-BORRAR-PEDIDO</u>.

	3. El flujo continuo en el paso 4.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El CAJERO selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01: Se agrega un nuevo pedido a la base de datos del sistema.
Incluye:	CU-21
Extiende:	N/A

Tabla 3 Descripción CU-03

3.3.3 Diagrama de robustez

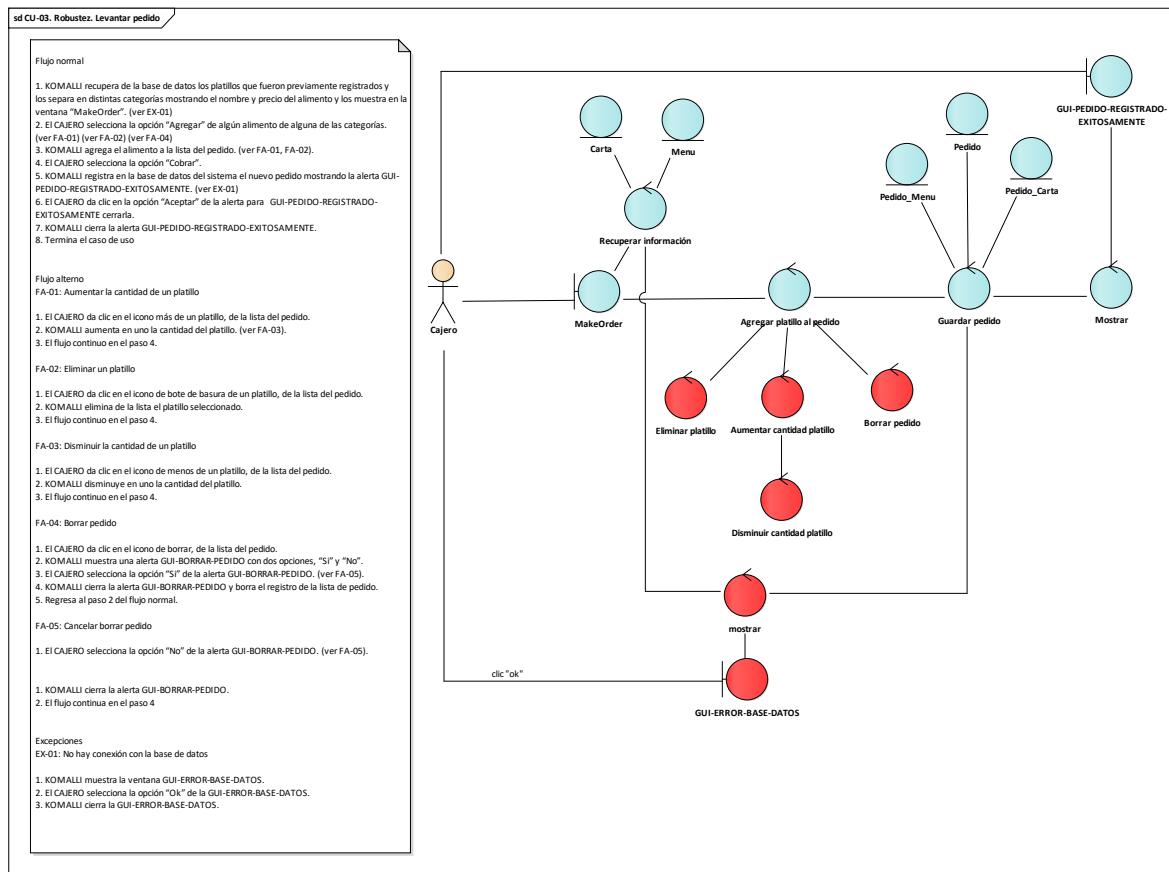


Ilustración 16 Diagrama de robustez CU-03

3.3.4 Diagrama de secuencia

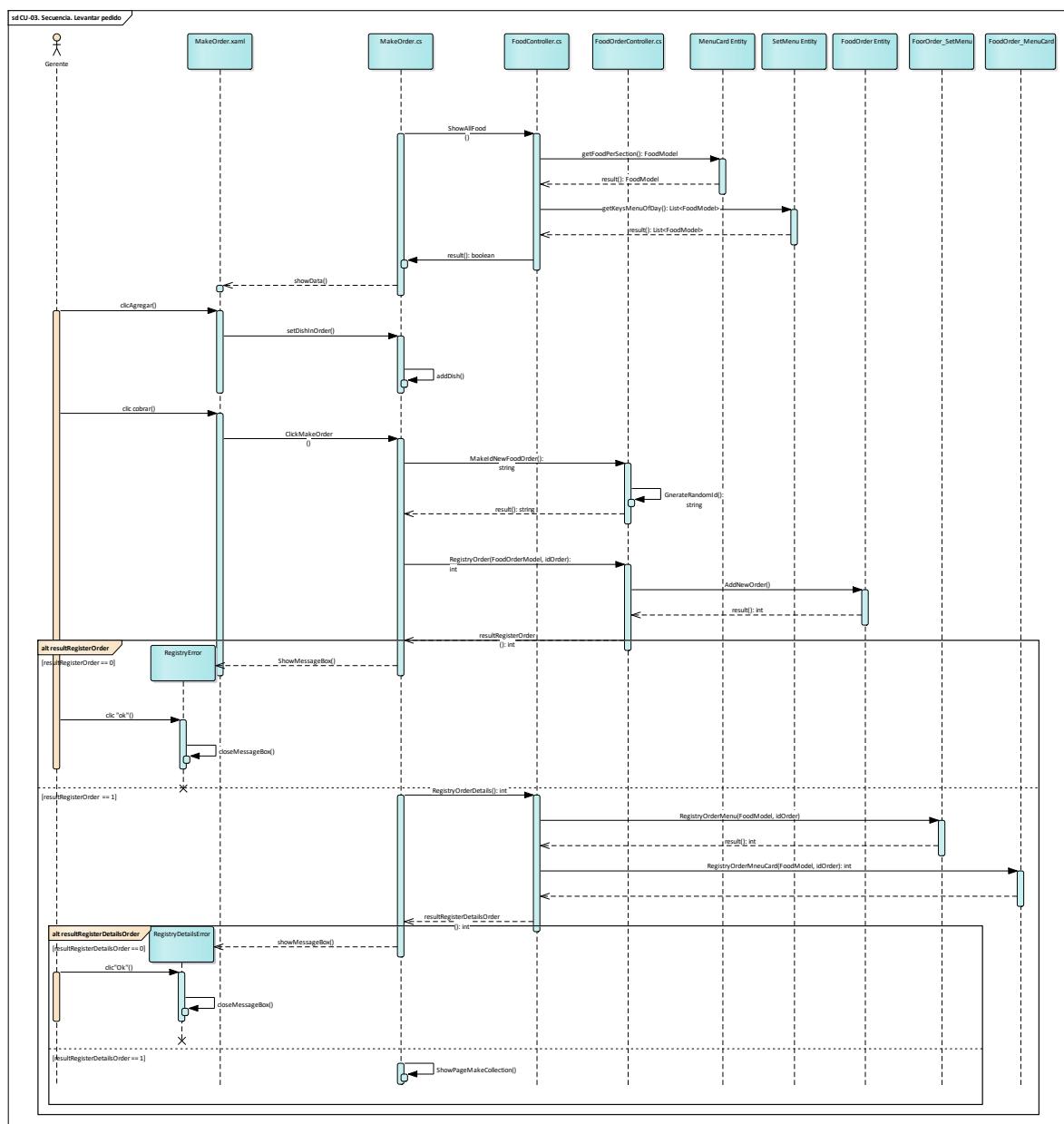


Ilustración 17 Diagrama de secuencia CU-03

3.4 CU 04 Consultar usuario

3.4.1 Prototipo

El prototipo muestra una interfaz web con un diseño moderno. En la parte superior izquierda, hay un logo de Komalli y un menú lateral con opciones como 'Usuarios' (seleccionada), 'Consultar' y 'Registrar'. La sección central tiene un encabezado 'Lista de usuarios registrados' y un campo de búsqueda. Abajo de esto, se encuentra una tabla que detalla los datos de cinco usuarios: Miriam Ramírez Zárate (Cajero, Activo), Paloma Osiris Báez (Personal de cocina, Inactivo), Alejandro Sánchez Marín (Personal de cocina, Activo), Ares Judda Rivera Soto (Cajero, Nuevo) y Rosa Ysela Lara Landa (Jefe de cocina, Activo). Cada fila de la tabla incluye tres puntos suspensivos y un botón 'Modificar'.

Ilustración 18 Prototipo CU-04

3.4.2 Descripción

ID:	CU- 04
Nombre del CU:	Consultar usuario
Requisito funcional:	RF-02: Komalli debe permitir la consulta así como la modificación de información registrada de los usuarios. RF-03: Se debe permitir la búsqueda de usuarios por nombre, lo cual debe arrojar un resultado de búsqueda ordenado.
Descripción:	Se consultan los datos de un usuario en la base de datos del sistema
Actor(es):	Gerente (Primario)
Disparador:	El Gerente da clic en “Consultar” de la sección Usuarios del menú del sistema.
Precondiciones:	PRE-01: Existe al menos un usuario registrado en el sistema.
Flujo Normal:	<ol style="list-style-type: none">KOMALLI recupera de la base de datos los datos de los usuarios registrados en el sistema, mostrando el numeroPersonal, nombre, rol y estatus en la ventana “ConsultUsers”. (ver EX01).El GERENTE ingresa un nombre dentro del buscador de la ventana “ConsultUsers” y da clic en ícono de buscar (una lupa).KOMALLI busca aquellos nombres de usuarios que coincidan con la palabra ingresada en el buscador y los muestra en forma de tabla en la ventana “ConsultUsers”.Termina el caso de uso.

Flujos Alternos:	N/A
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 5. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 6. El GERENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 7. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 8. Fin del caso de uso.
Postcondiciones:	N/A
Incluye:	N/A
Extiende:	CU-02. Modificar usuario

Tabla 4. Descripción CU-04

3.4.3 Diagrama de robustez

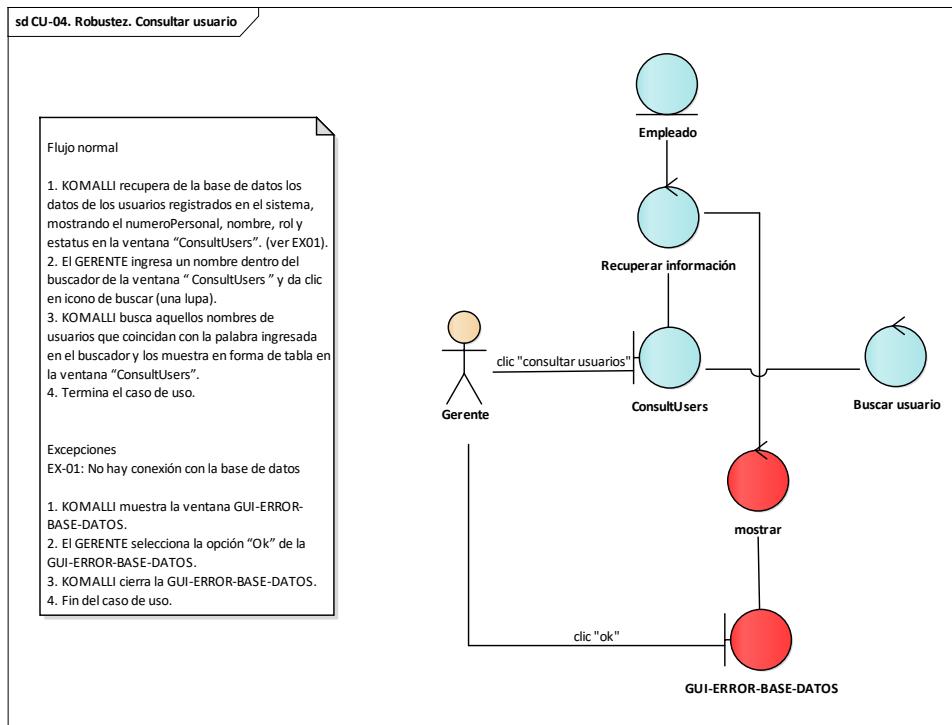


Ilustración 19 Diagrama de robustez CU-04

3.4.4 Diagrama de secuencia

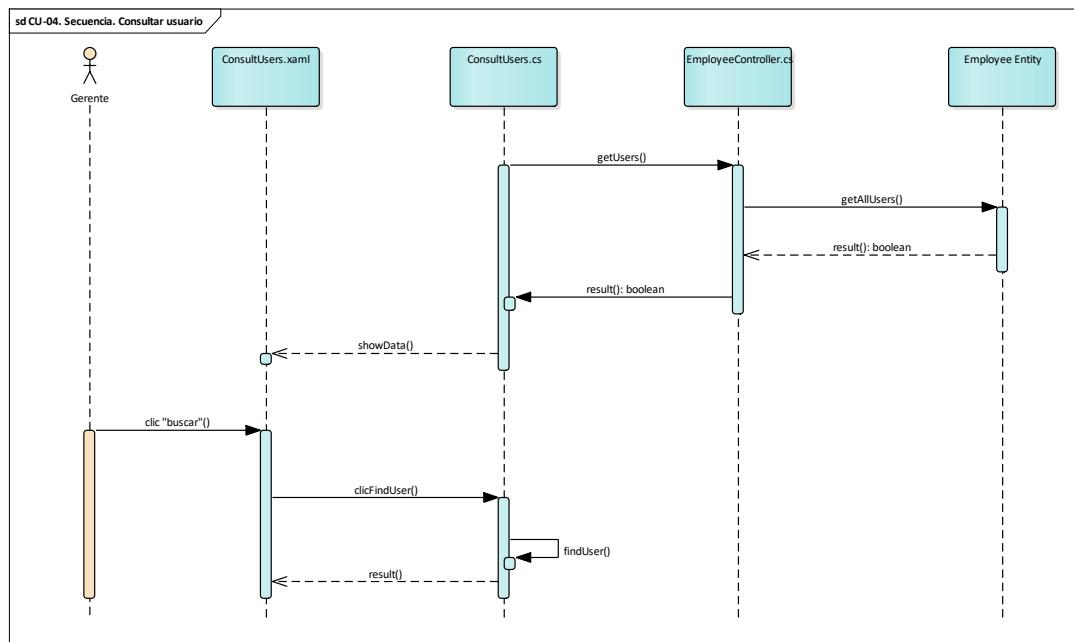


Ilustración 20 Diagrama de secuencia CU-04

3.5 CU 05 Agregar ingrediente a inventario

3.5.1 Prototipo

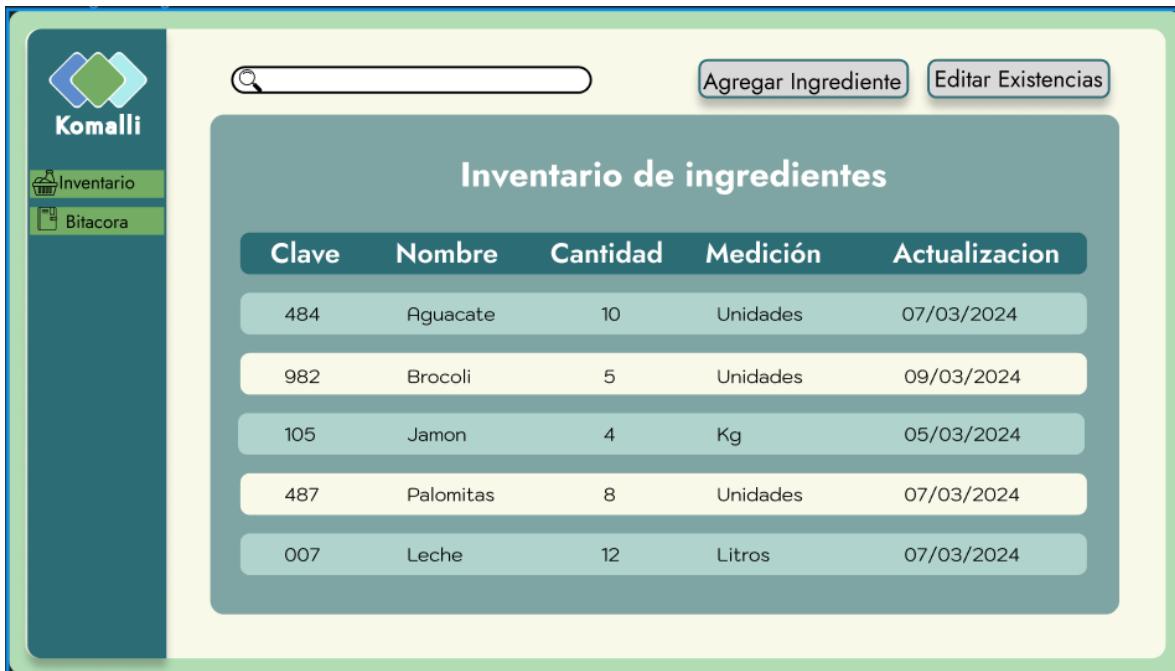


Ilustración 21 Prototipo CU-05

The image shows a mobile application interface for 'Komalli'. The top navigation bar includes a logo and the text 'Komalli' with two green diamond icons. Below the navigation, there are two tabs: 'Inventario' (Inventory) and 'Bitacora' (Log). The main content area has a teal header with the title 'Registro de ingredientes' (Ingredient Registration). It contains three input fields: 'Nombre de Ingrediente' (Ingredient Name), 'Cantidad' (Quantity), and 'Medición' (Measurement). A dropdown arrow is positioned below the 'Medición' field. At the bottom right is a button labeled 'Agregar Ingrediente' (Add Ingredient).

Ilustración 22 Prototipo CU-05

This image is identical to Ilustración 22, but it includes a small white rectangular callout box with a black border containing the text 'El ingrediente fue registrado' (The ingredient was registered) centered over the 'Cantidad' input field.

Ilustración 23 Prototipo CU-05

3.5.2 Descripción

ID:	CU- 5
Nombre del CU:	Registrar ingrediente en inventario
Requisito funcional:	RF-05: Se debe tener la posibilidad de registrar nuevos ingredientes detallando su nombre, código, descripción, precio, restricciones (si aplican), foto y cantidad disponible.
Descripción:	Se busca el registro de ingredientes dentro del inventario del comedor universitario para su control y seguimiento.
Actor(es):	Jefe Cocina (Primario)
Disparador:	El Jefe de Cocina da clic en Agregar Ingrediente dentro de la sección del inventario de ingredientes.
Precondiciones:	<p>PRE-01.- El usuario debe estar registrado en el sistema</p> <p>PRE-02.- El usuario debe de tener su perfil habilitado dentro del sistema.</p> <p>PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Jefe de cocina).</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI muestra la ventana “RegisterIngredient” con un formulario con los campos <i>nombre del ingrediente, cantidad y medición</i>. 2. El JEFE DE COCINA llena los campos solicitados por el formulario y da clic en “agregar ingrediente” de “RegisterIngredient”. 3. KOMALLI verifica y valida los datos del ingrediente ingresados en el formulario. (ver FA-01, ver FA-02). 4. KOMALLI guarda los datos en la base de datos y muestra la alerta <u>REGISTRO-INGREDIENTE-EXISTOSO</u>. (ver EX-01). 5. El JEFE DE COCINA da clic en “Aceptar” de la alerta <u>REGISTRO-INGREDIENTE-EXISTOSO</u>. 6. KOMALLI cierra la alerta <u>REGISTRO-INGREDIENTE-EXISTOSO</u>. 7. Finaliza el caso de uso.
Flujos Alternos:	<p>FA-01.- Datos inválidos</p> <ol style="list-style-type: none"> 1. KOMALLI detecta campos inválidos y muestra la alerta <u>DATOS-INVALIDOS</u>. 2. El JEFE DE COCINA da clic en la opción “Ok” de la alerta <u>DATOS-INVALIDOS</u>. 3. KOMALLI cierra alerta <u>DATOS-INVALIDOS</u>. 4. Regresa al paso 2 del flujo normal. <p>FA-02.- Ingrediente existente</p> <ol style="list-style-type: none"> 1. KOMALLI detecta que el ingrediente que se trata de registrar, ya se encuentra registrado y muestra la alerta <u>INGREDIENTE-EXISTENTE</u>. 2. El JEFE DE COCINA da clic en la opción “Ok” de la alerta <u>INGREDIENTE-EXISTENTE</u>. 3. KOMALLI cierra alerta <u>INGREDIENTE-EXISTENTE</u>. 4. Regresar al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>ERROR-BASE-DATOS</u>. 2. El JEFE DE COCINA selecciona la opción “Ok” de la <u>ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01.- Un nuevo ingrediente es registrado en la base de datos del sistema.

	POST-02.- El JEFE DE COCINA es redirigido a la ventana principal del inventario.
Incluye:	N/A
Extiende:	N/A

Tabla 5 Descripción CU-05

3.5.3 Diagrama de robustez

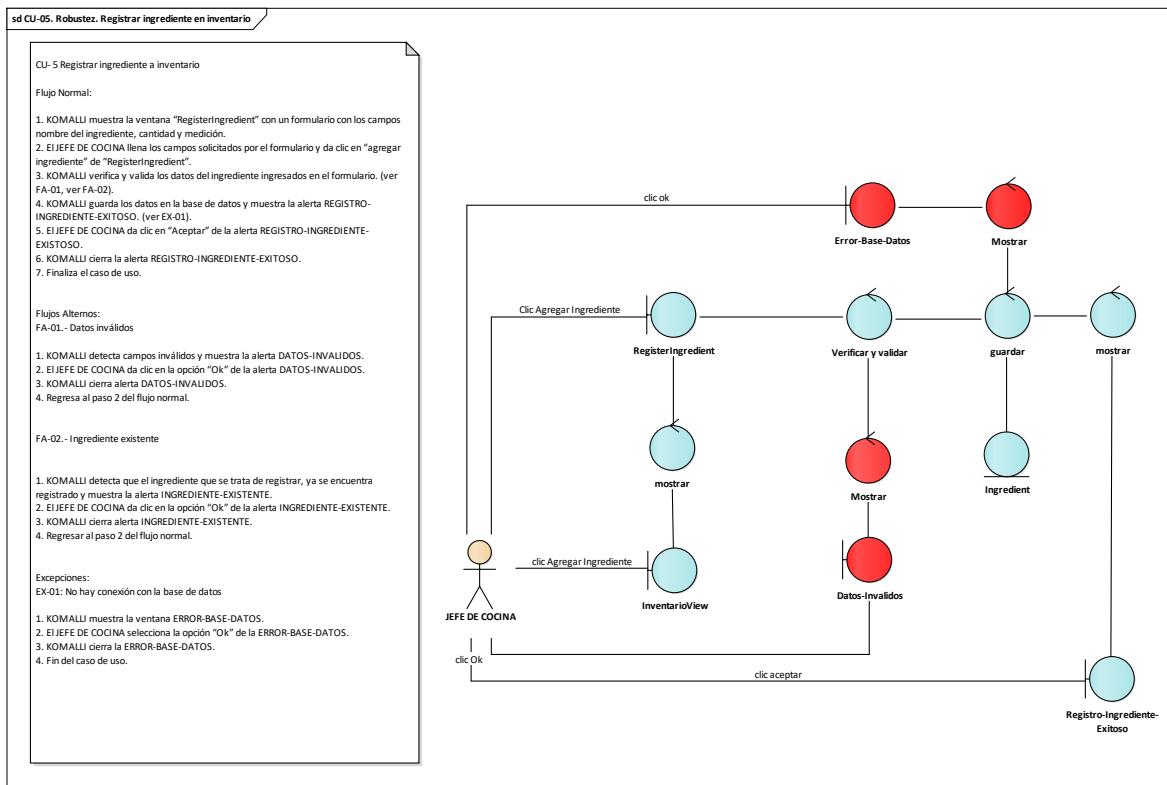


Ilustración 24 Diagrama de robustez CU-05

3.5.4 Diagrama de secuencia

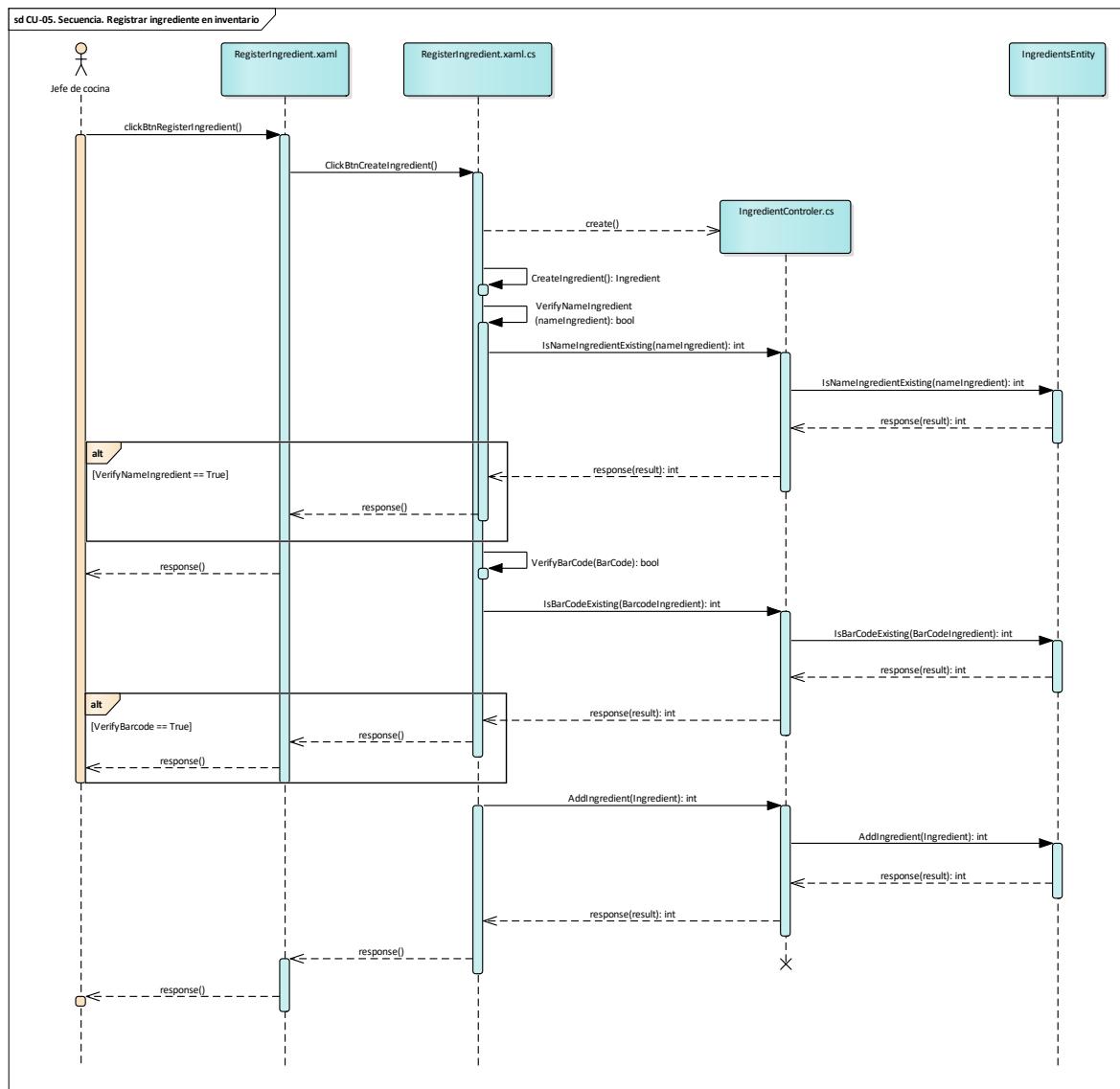


Ilustración 25 Diagrama de secuencia CU-05

3.6 CU 06 Modificar ingrediente

3.6.1 Prototipo

Este prototipo muestra la interfaz para modificar el inventario de ingredientes. En la parte superior izquierda, hay un sidebar con el logo Komalli y los menús "Inventario" y "Bitacora". La barra superior tiene un campo de búsqueda y un botón "Editar Existencias". El título central es "Inventario de ingredientes". La tabla de datos incluye columnas para Clave, Nombre, Cantidad, Medición y Fecha Reabasto. Los datos actuales son:

Clave	Nombre	Cantidad	Medición	Fecha Reabasto
484	Aguacate	11	Unidades	07/03/2024
982	Brocoli	0	Unidades	09/03/2024
105	Jamon	4	Kg	05/03/2024
487	Palomitas	2	Unidades	07/03/2024
007	Leche	12	Litros	07/03/2024

En la parte inferior derecha, hay un botón "Guardar cambios".

Ilustración 26 Prototipo CU-06

Este prototipo muestra la misma interfaz que el anterior, pero con una notificación emergente en el centro de la pantalla. La notificación dice: "Las existencias en el inventario de ingredientes han sido actualizadas".

Ilustración 27 Prototipo CU-06

3.6.2 Descripción

ID:	CU- 6
Nombre del CU:	Editar inventario
Requisito funcional:	RF-06: Debe permitir la edición de la información de ingredientes registrados, excepto el código.
Descripción:	Se busca la modificación de la existencia de los ingredientes del comedor para su seguimiento y posteriores operaciones de reabastecimiento.
Actor(es):	Jefe Cocina (Primario)
Disparador:	El Jefe de Cocina da clic en Modificar existencias dentro de la sección del inventario de ingredientes.
Precondiciones:	<p>PRE-01.- El usuario debe estar registrado en el sistema</p> <p>PRE-02.- El usuario debe de tener su perfil habilitado dentro del sistema.</p> <p>PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Jefe de cocina).</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI en la ventana “inventarioIngredientes” habilita la edición en el inventario solamente en la columna de <i>cantidad</i>. 2. El JEFE DE COCINA modifica la <i>cantidad</i> existente hasta el momento de los ingredientes del comedor y da clic en guardar cambios. (Ver FA-01) 3. KOMALLI verifica y valida la/s cantidades de ingrediente/s ingresadas. (ver FA-02). 4. KOMALLI actualiza los datos en la base de datos y muestra la alerta <u>ACTUALIZACION-INGREDIENTE-EXISTOSO</u>. (ver EX-01). 5. El JEFE DE COCINA da clic en “Aceptar” de la alerta <u>ACTUALIZACION-INGREDIENTE-EXISTOSO</u>. 6. KOMALLI cierra la alerta <u>ACTUALIZACION-INGREDIENTE-EXISTOSO</u>. 7. Finaliza el caso de uso.
Flujos Alternos:	<p>FA-01.- Buscar ingrediente.</p> <ol style="list-style-type: none"> 1. El JEFE DE COCINA busca el ingrediente que desee cambiar su existencia en la barra de búsqueda, ya sea por su <i>nombre</i>, <i>clave</i> o <i>código de barras</i> (si aplica). 2. KOMALLI muestra los resultados de la búsqueda de ingredientes que hagan match con lo solicitado. 3. Regresa al paso 2 del flujo normal. <p>FA-02.- Datos inválidos</p> <ol style="list-style-type: none"> 1. KOMALLI detecta campos inválidos y muestra la alerta <u>DATOS-INVALIDOS</u>. 2. El JEFE DE COCINA da clic en la opción “Ok” de la alerta <u>DATOS-INVALIDOS</u>. 3. KOMALLI cierra alerta <u>DATOS-INVALIDOS</u>. 4. Regresa al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>ERROR-BASE-DATOS</u>. 2. El JEFE DE COCINA selecciona la opción “Ok” de la <u>ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	<p>POST-01.- La <i>cantidad</i> de existencia de uno o más ingredientes es registrado en la base de datos del sistema.</p> <p>POST-02.- El JEFE DE COCINA es redirigido a la ventana principal del inventario.</p>
Incluye:	N/A

Extiende:	N/A
-----------	-----

Tabla 6 Descripción CU-06

3.6.3 Diagrama de robustez

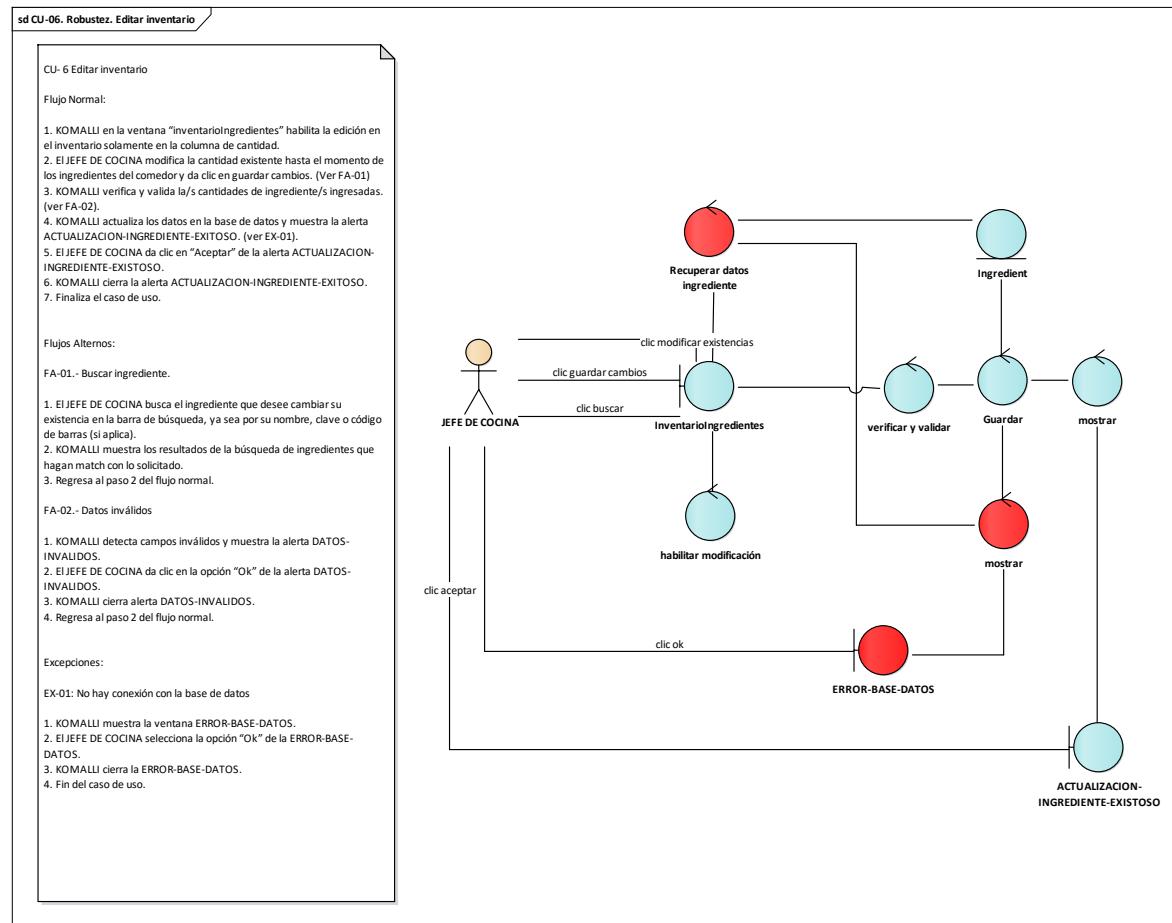


Ilustración 28 Diagrama de robustez CU-06

3.6.4 Diagrama de secuencia

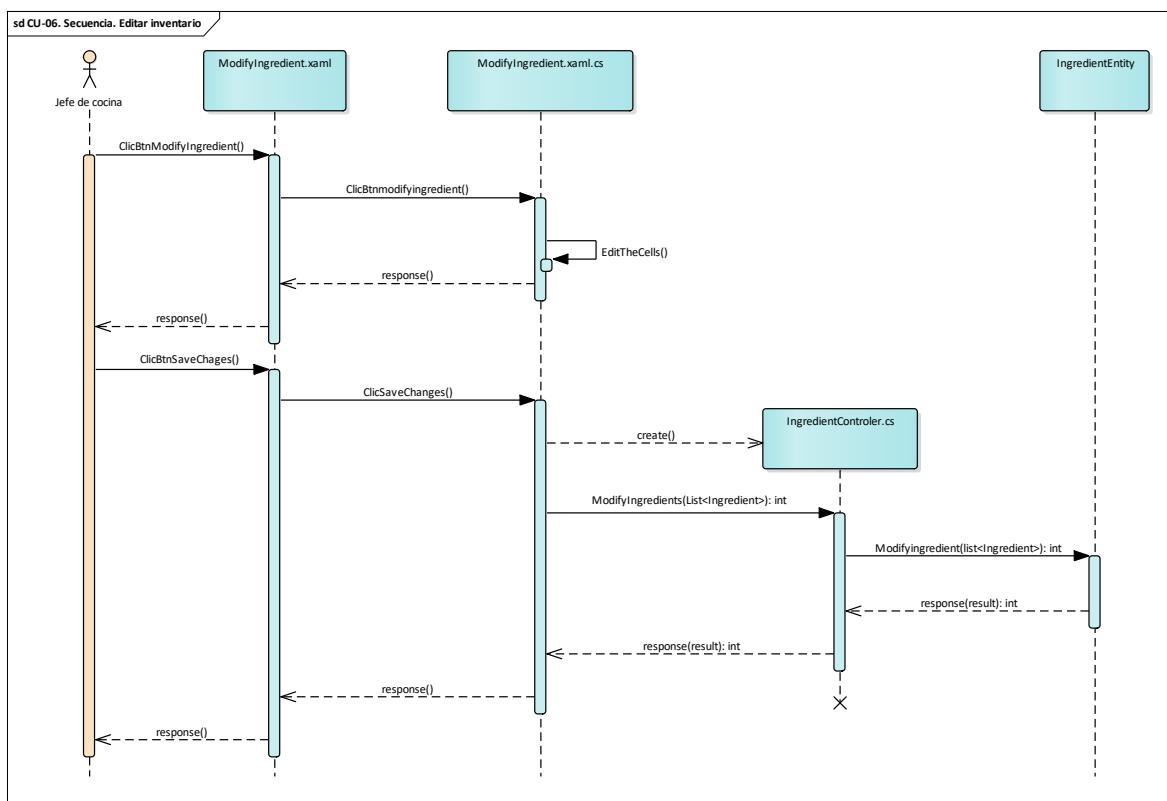


Ilustración 29 Diagrama de secuencia CU-06

3.7 CU 07 Buscar ingrediente en inventario

3.7.1 Prototipo

Ilustración 30 Prototipo CU-07

Este prototipo muestra la interfaz para buscar un ingrediente en el inventario. En la parte superior izquierda, hay un logo de Komalli y una barra lateral con los íconos de 'Inventario' y 'Bitácora'. En la parte superior central, hay un campo de búsqueda con un icono de lupa y dos botones: 'Agregar Ingrediente' y 'Editar Existencias'. El título 'Inventario de ingredientes' se encuentra en el centro superior de la pantalla. Una tabla detallada muestra los siguientes datos:

Clave	Nombre	Cantidad	Medición	Fecha Reabasto
484	Aguacate	10	Unidades	07/03/2024
982	Brocoli	5	Unidades	09/03/2024
105	Jamon	4	Kg	05/03/2024
487	Palomitas	8	Unidades	07/03/2024
007	Leche	12	Litros	07/03/2024

Ilustración 31 Prototipo CU-07

Este prototipo muestra la misma interfaz que la Ilustración 30, pero con un resultado de búsqueda específico. La barra de búsqueda contiene la palabra 'Agua'. La tabla muestra los siguientes resultados:

Clave	Nombre	Cantidad	Medición	Fecha Reabasto
484	Aguacate	10	Unidades	07/03/2024
1082	Agua en botellada	20	Unidades	09/03/2024
1050	Agua mineral	4	Unidades	05/03/2024

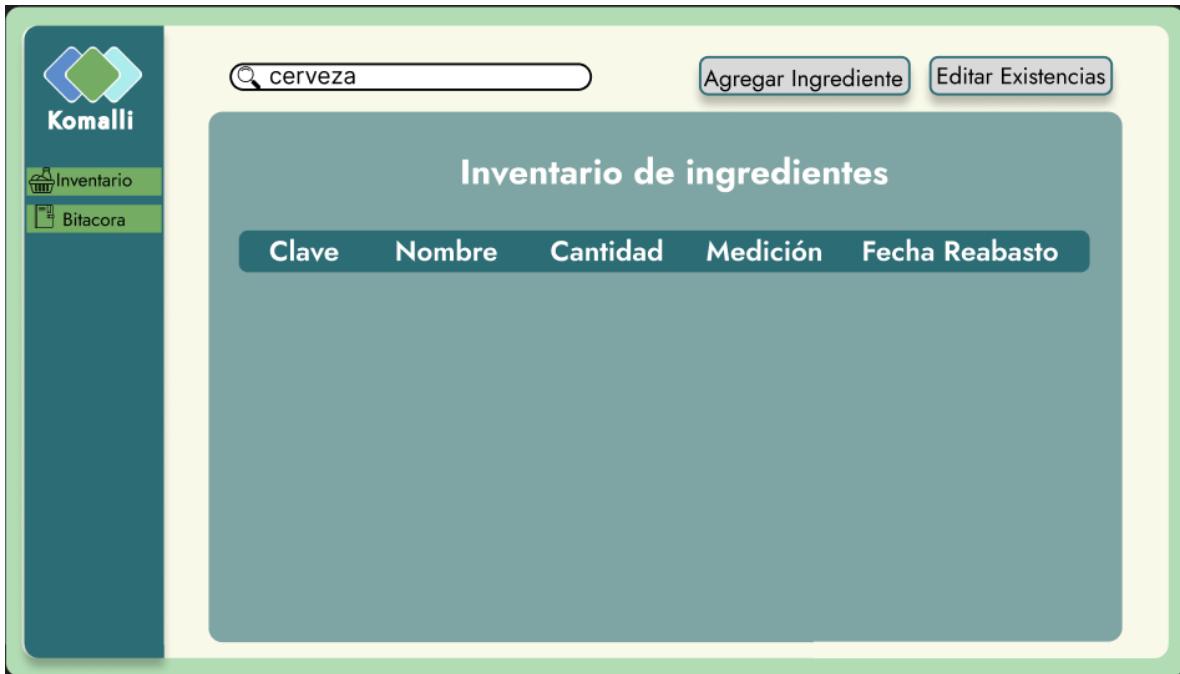


Ilustración 32 Prototipo CU-07

3.7.2 Descripción

ID:	CU-07
Nombre del CU:	Buscar ingrediente en inventario
Requisito funcional:	RF-07: Komalli debe permitir a los empleados del comedor universitario buscar ingredientes por nombre o código, y mostrar los resultados de forma ordenada.
Descripción:	Se busca que el jefe de cocina pueda acceder a la información de ciertos ingredientes del inventario de su interés, ya sea por nombre, código de barras o clave de este.
Actor(es):	Jefe Cocina (Primario)
Disparador:	El jefe de Cocina escribe sobre la barra de búsqueda ubicada sobre el inventario de ingredientes
Precondiciones:	<p>PRE-01.- El usuario debe estar registrado en el sistema</p> <p>PRE-02.- El usuario debe de tener su perfil habilitado dentro del sistema.</p> <p>PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Jefe de cocina).</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI en la ventana “inventarioIngredientes” con una vista general del inventario y una barra de búsqueda. 2. El JEFE DE COCINA escribe sobre la barra de búsqueda de ingredientes. 3. KOMALLI hace match con los caracteres ingresados en la barra de búsqueda, mostrando aquellos ingredientes que tengan dichos caracteres en su <i>nombre, clave o código de barras</i>. (VER EX-01) 4. KOMALLI muestra los ingredientes que coincidan con la búsqueda. (VER FA-01) 5. Finaliza el caso de uso.
Flujos Alternos:	<p>FA-01.- Ingrediente inexistente.</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la tabla de resultados vacía, lo que se interpreta como que el sistema no ha encontrado ningún elemento que coincida con la búsqueda solicitada; ya sea por <i>nombre, clave o código de barras</i> 2. Regresa al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>ERROR-BASE-DATOS</u>. 2. El JEFE DE COCINA selecciona la opción “Ok” de la <u>ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01.- KOMALLI muestra la información general de los ingredientes que coincidan con la búsqueda: <i>nombre, clave, cantidad, medición, fecha de actualización</i>
Incluye:	N/A
Extiende:	N/A

Tabla 7. Descripción CU-07

3.7.3 Diagrama de robustez

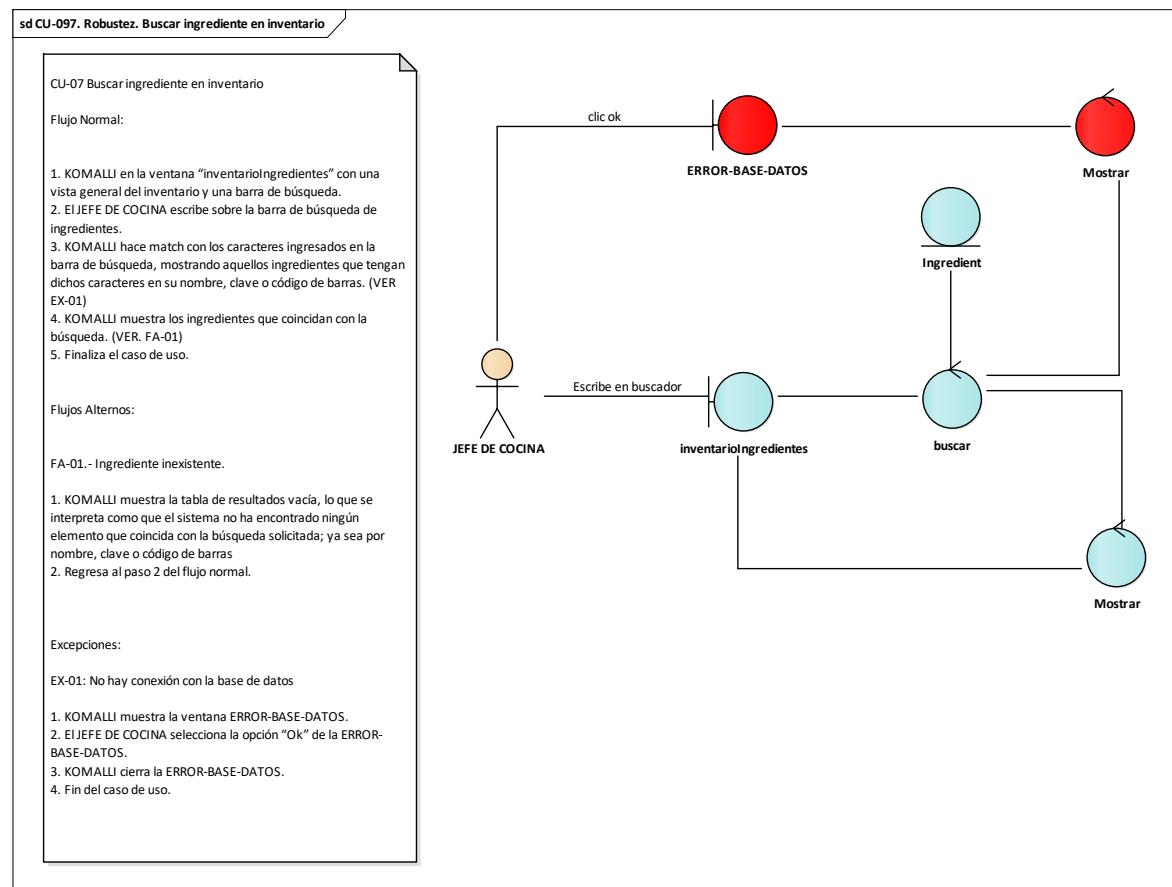


Ilustración 33 Diagrama de robustez CU-07

3.7.4 Diagrama de secuencia

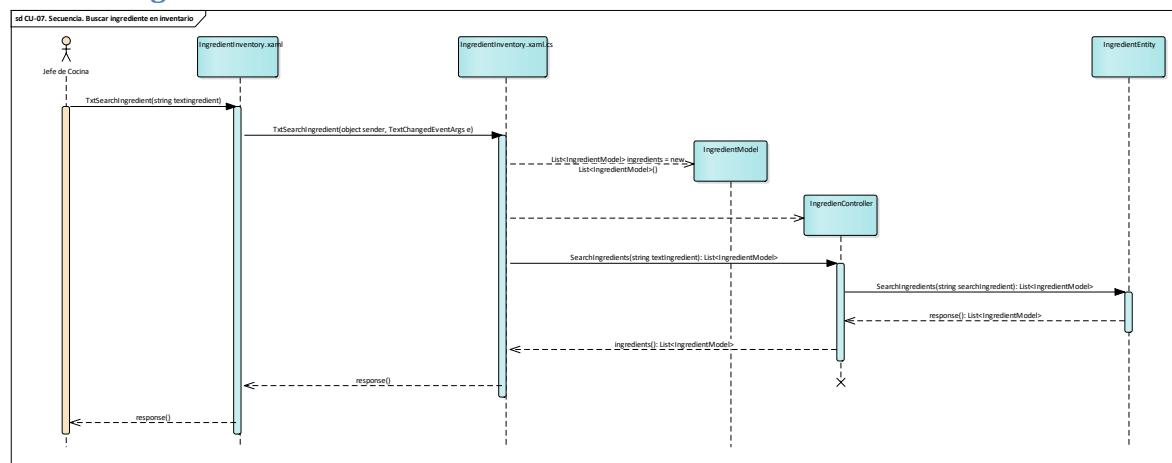


Ilustración 34 Diagrama de secuencia CU-07

3.8 CU 08 Generar reporte de Inventario

3.8.1 Prototipo

El prototipo muestra la interfaz de usuario de la aplicación Komalli. En el lado izquierdo, hay un sidebar con el logo Komalli y una lista de opciones: Usuarios, Reportes (que incluye Inventario, Bitácora y Movimientos diarios), y un apartado personal que dice Miriam Ramírez Zárate.

En el centro, se visualiza un cuadro titulado "Reporte de inventario" que contiene una tabla con los siguientes datos:

Clave	Nombre	Cantidad	Medición
484	Aguacate	10	Unidades
982	Brocoli	5	Unidades
105	Jamón	4	Kg
487	Palomitas	8	Unidades

En la parte superior derecha del cuadro central, hay un botón "Descargar".

Ilustración 35 Prototipo CU-08

3.8.2 Descripción

ID:	CU-08
Nombre del CU:	Generar reporte de inventario
Requisito funcional:	RF-08: Se deben generar informes de inventario en formato PDF.
Descripción:	Se crea un reporte con la información resumida de las últimas altas en el inventariado de cocina.
Actor(es):	Gerente (Primario)
Disparador:	El Gerente da clic en el ícono de reporte para ingresar a “ IngredientReport ”.
Precondiciones:	PRE-01: Existe registrado en el sistema al menos un ingrediente .
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI recupera los ingredientes registrados en la base de datos y muestra la ventana “IngredientReport” con una opción crear el reporte. (ver EX-01) 2. KOMALLI muestra una previsualización del reporte. 3. El GERENTE selecciona una <i>categoria</i> del combobox y la opción “Descargar” de “GenerateReport”. (ver FA-01) 4. KOMALLI guarda el reporte en el dispositivo y muestra la <u>GUI-REPORTE-GUARDADO</u>. 5. El GERENTE selecciona la opción “Continuar” de la <u>GUI-REPORTE-GUARDADO</u>. 6. KOMALLI cierra la <u>GUI-REPORTE-GUARDADO</u>. 7. Termina el caso de uso.
Flujos Alternos:	N/A
Excepciones:	EX-01: No hay conexión con la base de datos <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El GERENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01: Un nuevo reporte es generado y guardado en el dispositivo del gerente .
Incluye:	N/A
Extiende:	N/A

Tabla 8. Descripción CU-08

3.8.3 Diagrama de robustez

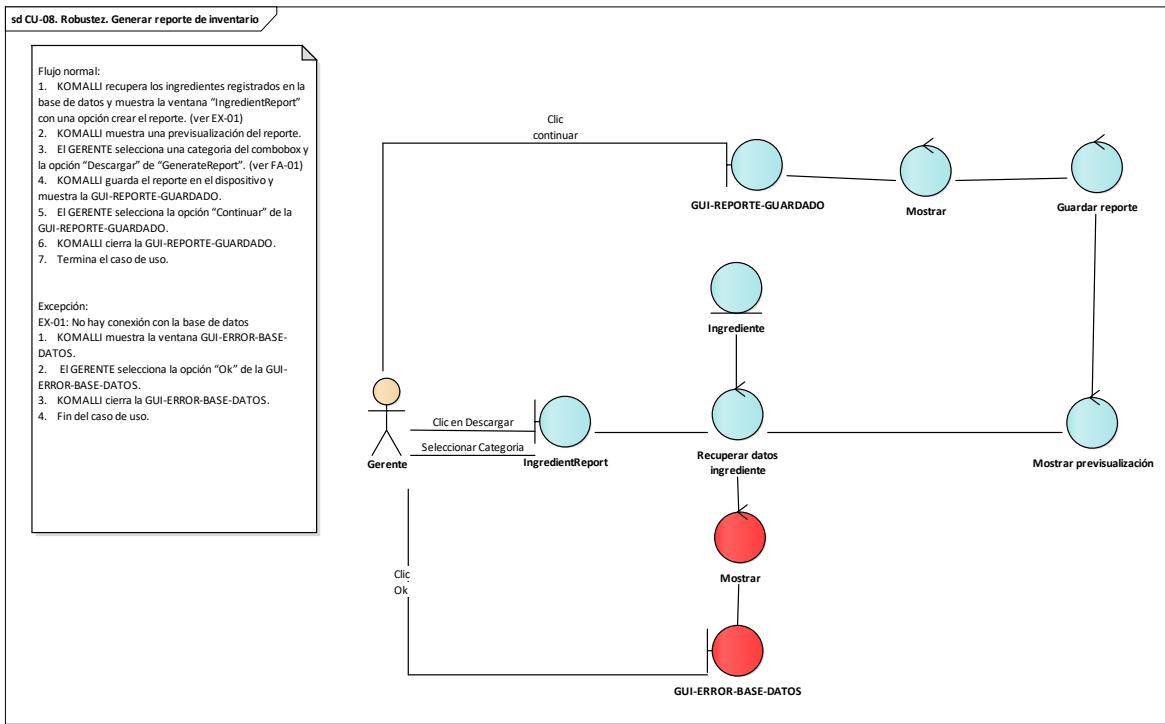


Ilustración 36 Diagrama de robustez CU-08

3.8.4 Diagrama de secuencia

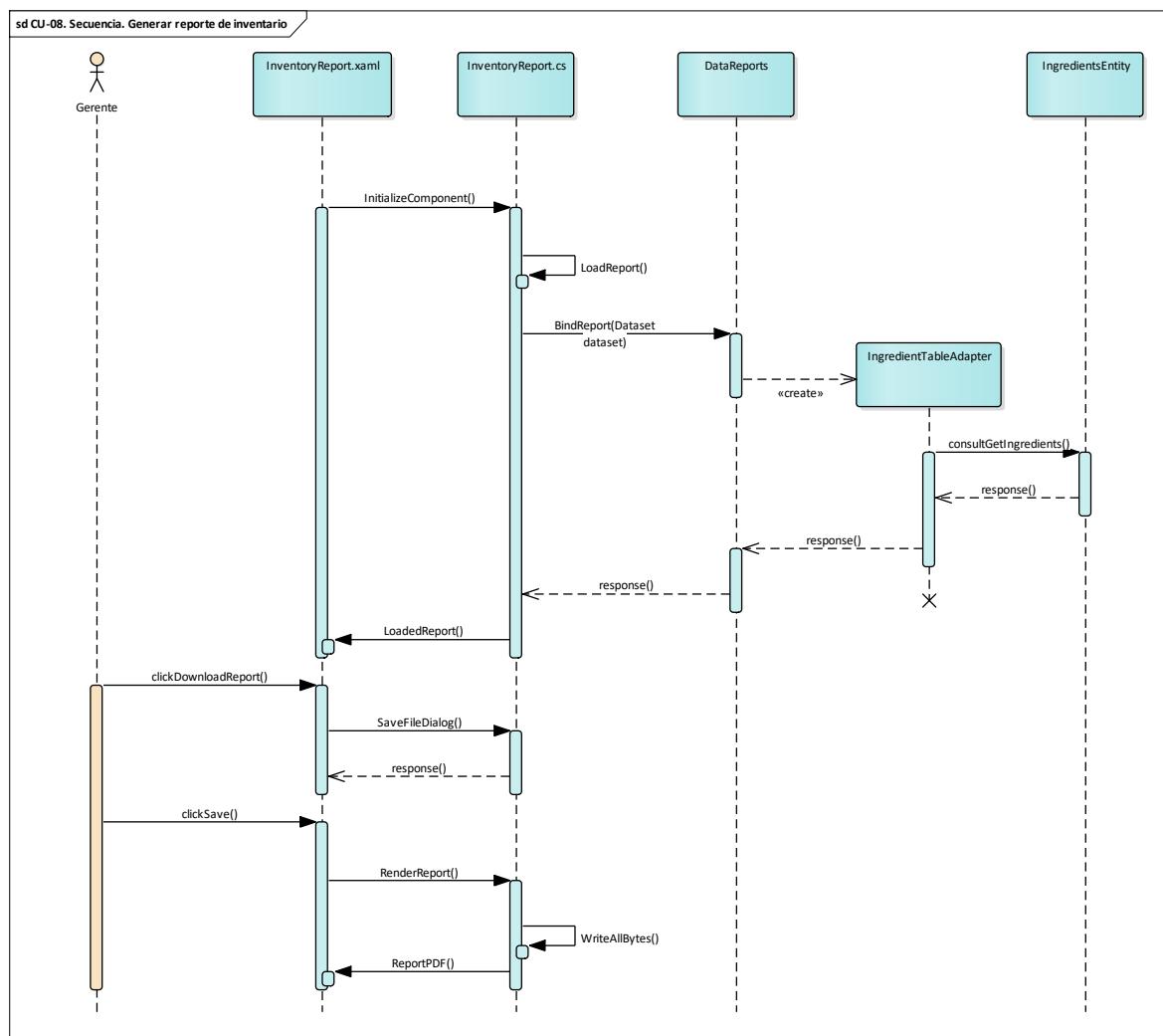


Ilustración 37 Diagrama de secuencia CU-08

3.9 CU 09 Genera pedido

3.9.1 Prototipo

The image shows a user interface for generating a meal order. On the left, a sidebar displays the Komalli logo and a summary of the meal: "Menu Desayuno" with a quantity of "1" and a total cost of "\$30". Below this are two buttons: "Cancelar" and "Continuar". On the right, a main panel titled "Datos de Pedido" contains fields for "Cantidad" (1), "Categoría" (Menú del día), "Tipo" (Desayuno), and "Costo" (\$30). Below these fields is a "Nombre:" input field with a placeholder and a "Confirmar" button.

Ilustración 38 Prototipo CU-09

This image is identical to Illustration 38, but the "Nombre:" input field now contains the name "Mildred". The rest of the interface, including the sidebar and the main data entry panel, remains the same.

Ilustración 39 Prototipo CU-09



Ilustración 40 Prototipo CU-09

3.9.2 Descripción

ID:	CU-09
Nombre del CU:	Generar pedido
Requisito Funcional:	RF-10: El sistema debe permitir a los clientes realizar pedidos especificando la fecha, productos y cantidades deseadas, así como el usuario solicitante y el estado del pedido.
Descripción:	Permite agregar alimento disponibles en menú de carta o menú del día para así poder realizar un pedido.
Actor(es):	Cliente (Primario)
Disparador:	El cliente selecciona una sección de alimento de su agrado.
Precondiciones:	PRE-01: Haber seleccionado un alimento.
Flujo Normal:	<p>1. KOMALLI muestra la pantalla “GenerateOrder” con los datos del pedido que incluye: cantidad, categoría, tipo, costo, junto con el total del pedido y campos que pide para llenar con información del cliente: nombre, matrícula o número de personal más un botón para “confirmar”.</p> <p>2. El CLIENTE ingresa la información solicitada en los campos correspondientes dando clic en el botón “confirmar”.</p> <p>3. KOMALLI valida que lo ingresado corresponda a los campos solicitados, guarda el pedido dentro de la base de datos. (FA-01) (EX01)</p> <p>4. KOMALLI muestra un mensaje sobre el pedido generado correctamente.</p> <p>5. Fin de caso de uso</p>
Flujos Alternos:	<p>FA-01: información de campos inválida.</p> <ol style="list-style-type: none"> 1. KOMALLI no acepta la información registrada dentro de los campos de texto. 2. El CLIENTE ingresa nuevamente la información solicitada. 3. Regreso al flujo normal número 3.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El CLIENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01: Un nuevo pedido se registra dentro de la base de datos.
Incluye:	N/A
Extiende:	N/A

Tabla 9 Descripción CU-09

3.9.3 Diagrama de robustez

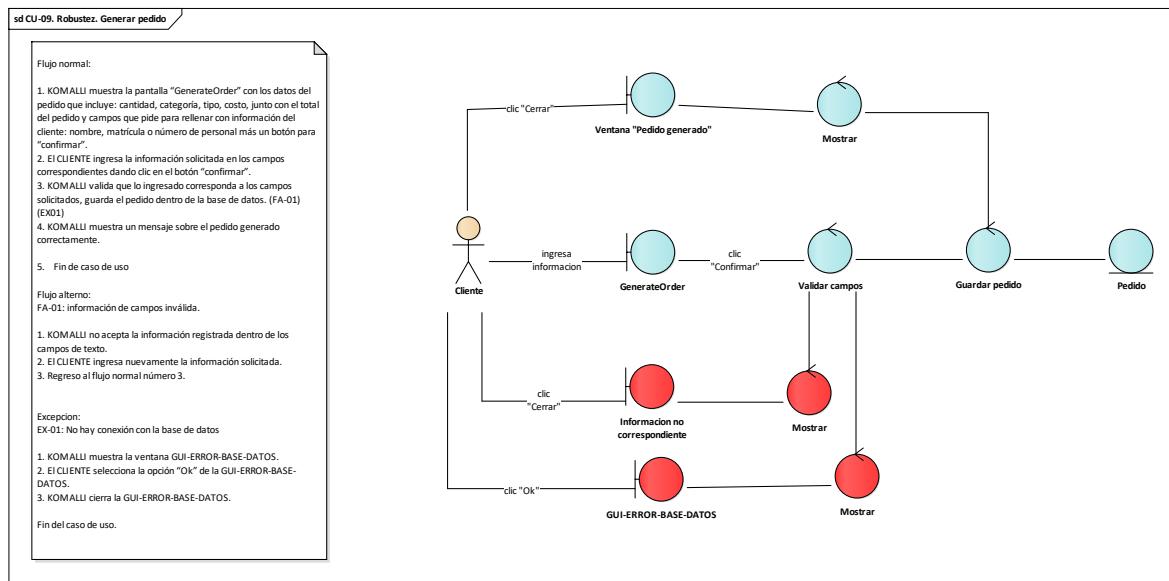


Ilustración 41 Diagrama de robustez CU-09

3.9.4 Diagrama de secuencia

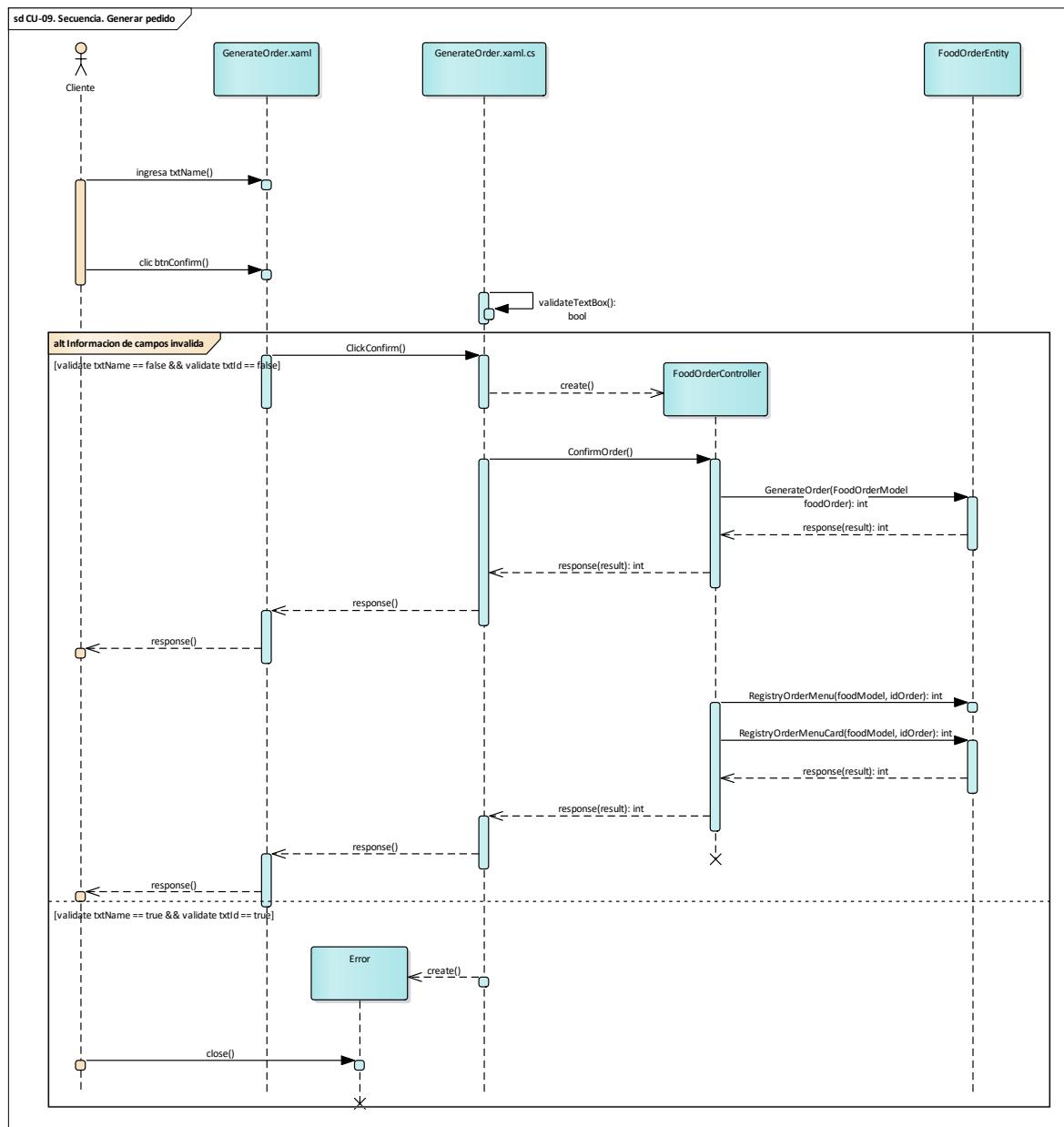


Ilustración 42 Diagrama de secuencia CU-09

3.10 CU 10 Editar pedido

3.10.1 Prototipo

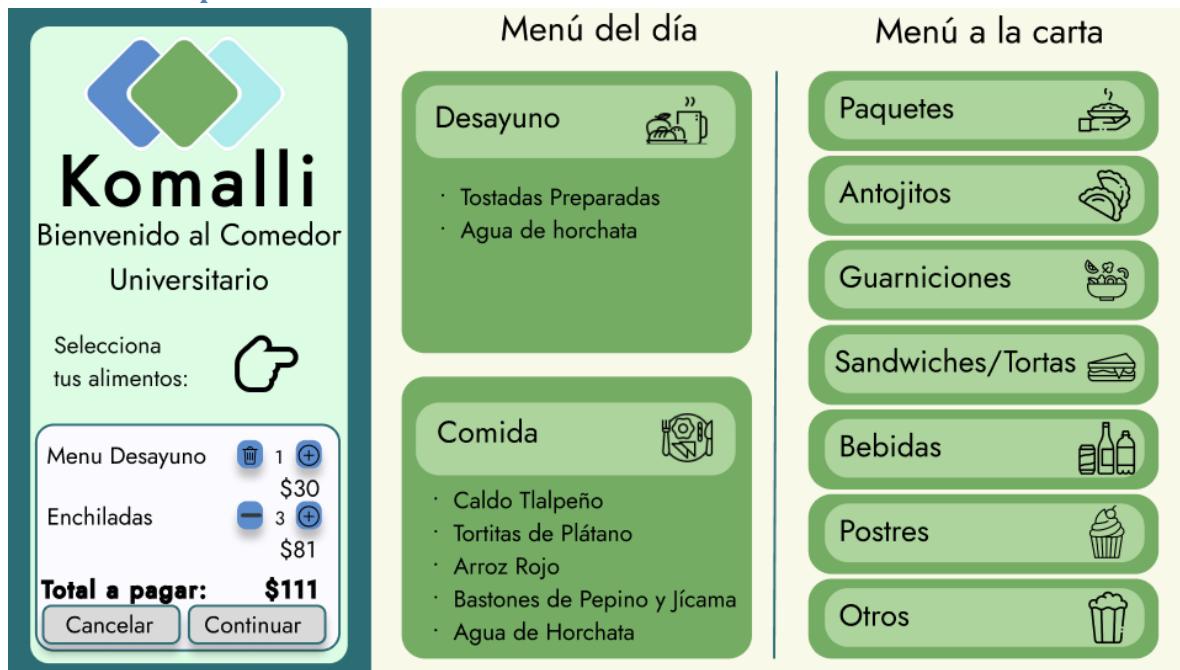


Ilustración 43 Prototipo CU-10

3.10.2 Descripción

ID:	CU-10
Nombre del CU:	Editar pedido
Requisito funcional:	RF-11: El sistema debe permitir a los usuarios editar pedidos existentes, agregar o eliminar productos y modificar cantidades.
Descripción:	Se suma, borra, o disminuye un pedido realizado por un cliente
Actor(es):	Cliente (Primario)
Disparador:	El cliente desea editar su pedido.
Precondiciones:	PRE-01: El cliente agrega por lo menos un alimento.
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI recupera el pedido realizado por el cliente mostrándole su alimento, el subtotal, así como dos botones para agregar o borrar los alimentos. 2. El CLIENTE selecciona el botón para agregar un alimento extra a su pedido. (ver FA-01) 3. KOMALLI cambia el ícono del botón borrar por el de disminuir y aumenta el subtotal del pedido. 4. El CLIENTE selecciona la opción “Continuar” 5. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Editar un alimento</p> <ol style="list-style-type: none"> 1. KOMALLI detecta que un alimento tiene como cantidad de 1. 2. El CLIENTE selecciona la opción “Borrar”. 3. KOMALLI elimina el alimento de la lista de pedidos y baja el subtotal del pedido. 4. Continua en el flujo normal 4.
Excepciones:	N/A
Postcondiciones:	POST-02: Un alimento es editado modificando la cantidad pedida. POST-03: El subtotal del pedido aumenta o disminuye.
Incluye:	N/A
Extiende:	N/A

Tabla 10 Descripción CU-10

3.10.3 Diagrama de robustez

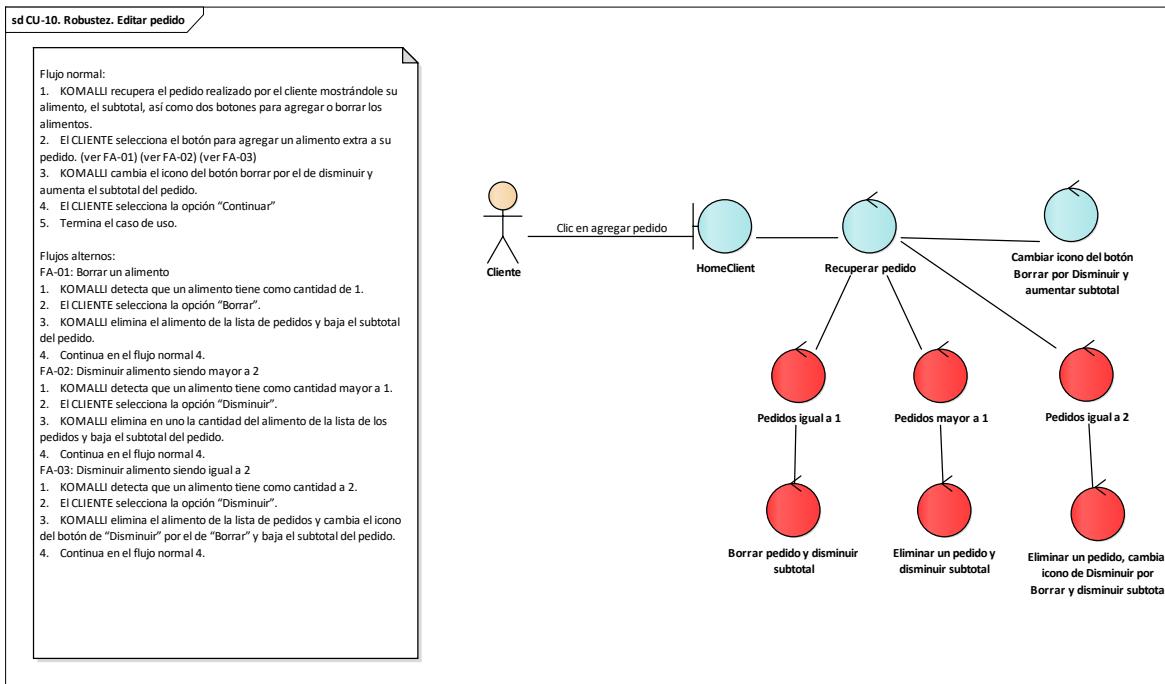


Ilustración 44 Diagrama de robustez CU-10

3.10.4 Diagrama de secuencia

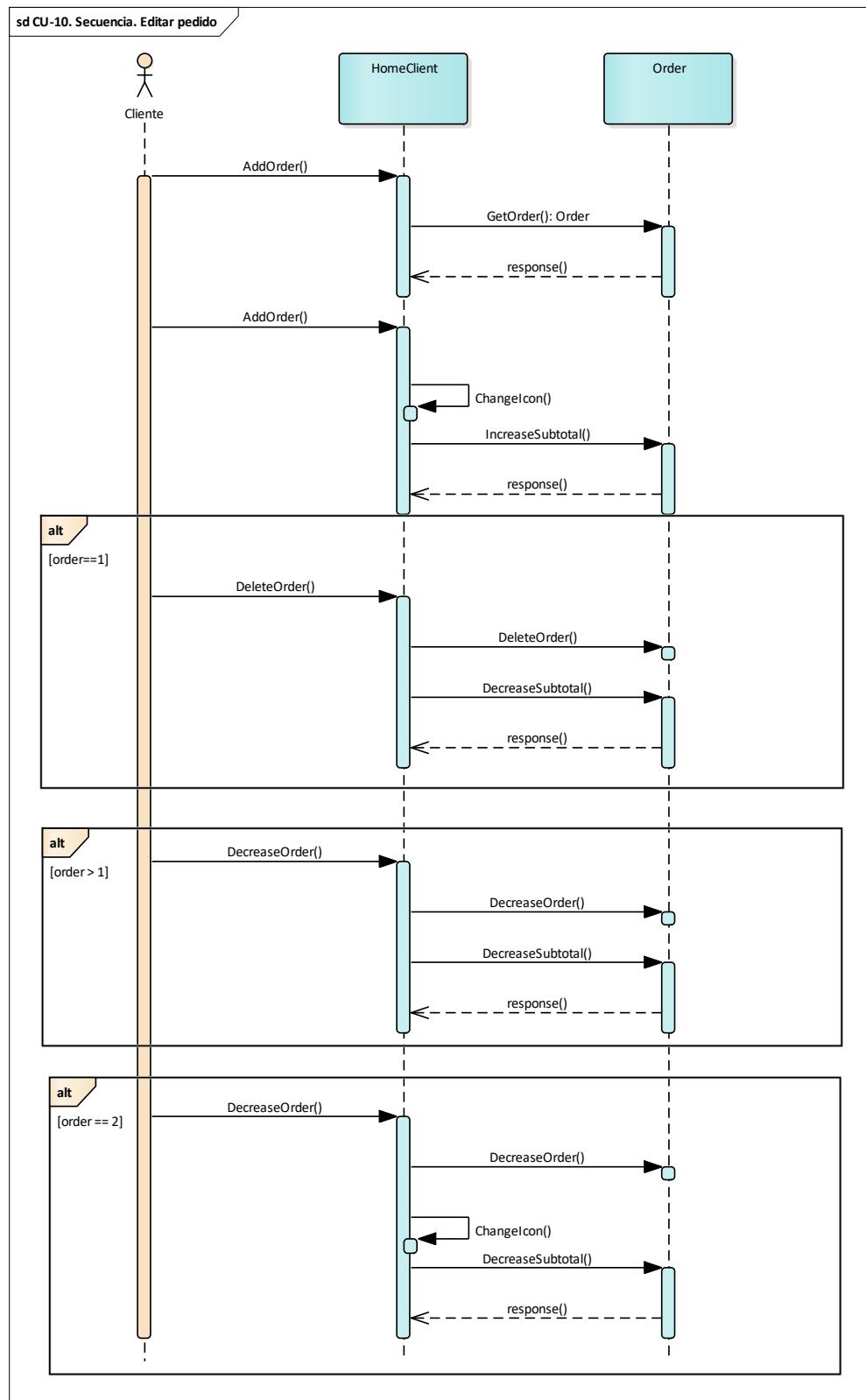


Ilustración 45 Diagrama de secuencia CU-10

3.11 CU 11 Editar Estado de pedido

3.11.1 Prototipo

El prototipo muestra la interfaz de usuario de la aplicación Komalli para la gestión de pedidos. En la parte superior izquierda, se encuentra el logo de Komalli y un menú lateral con tres opciones: Pedidos, Menú y Bitácora. La opción "Pedidos" está resaltada. En la parte superior central, hay un encabezado "Pedidos" y un filtro "Todo". La lista de pedidos se divide en cinco ítems, cada uno representado por una tarjeta:

Cantidad:	Categoría:	Tipo:	Estado:
1	Menú a la carta	Chilaquiles verdes	Hecho
2	Menú del día	Comida	Pendiente
1	Menú a la carta	Club sandwich	Pendiente
3	Menú del día	Comida	Pendiente
1	Menú a la carta	Enmoladas	Pendiente

Ilustración 46 Prototipo CU-11

3.11.2 Descripción

ID:	CU-11
Nombre del CU:	Editar estado de pedido
Requisito Funcional:	RF-14: Debe permitir a los empleados visualizar la lista completa de pedidos a preparar, incluyendo detalles como fecha del pedido, nombre del usuario, productos solicitados y cantidad requerida.
Descripción:	Permite al personal comida marcar como realizado un pedido dentro del sistema
Actor(es):	Personal Comedor (Primario)
Disparador:	El PERSONAL COMIDA da clic en pedidos dentro del menú de KOMALLI
Precondiciones:	PRE-01: Debe existir por lo menos un pedido generado dentro de KOMALLI
Flujo Normal:	<p>1. KOMALLI muestra la pantalla de “Orders” mostrando la información del pedido que incluye <i>cantidad</i>, <i>categoría</i>, <i>tipo</i>, más un botón “Pendiente”.</p> <p>2. El PERSONAL COMEDOR da clic en el botón para marcar el pedido como “Hecho”. (FA-01)</p> <p>3. KOMALLI muestra el pedido como “hecho” dentro de la pantalla.</p> <p>4. KOMALLI guarda el estado del pedido como hecho dentro de la base de datos. (EX01)</p> <p>5. Fin de caso de uso.</p>
Flujos Alternos:	<p>FA-01: Marcar pedido como Entregado</p> <ol style="list-style-type: none"> 1. El PERSONAL COMEDOR da clic en el botón “Hecho” 2. KOMALLI muestra el pedido como “entregado” dentro de la pantalla. 3. KOMALLI guarda el estado del pedido como Entregado dentro de la base de datos. (EX01) 4. Fin de caso de uso.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 5. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 6. El CLIENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 7. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 8. Fin del caso de uso.
Postcondiciones:	POST-01: Se ha actualizado el estado de un pedido dentro de la base de datos.
Incluye:	N/A
Extiende:	N/A

Tabla 11 Descripción CU-11

3.11.3 Diagrama de robustez

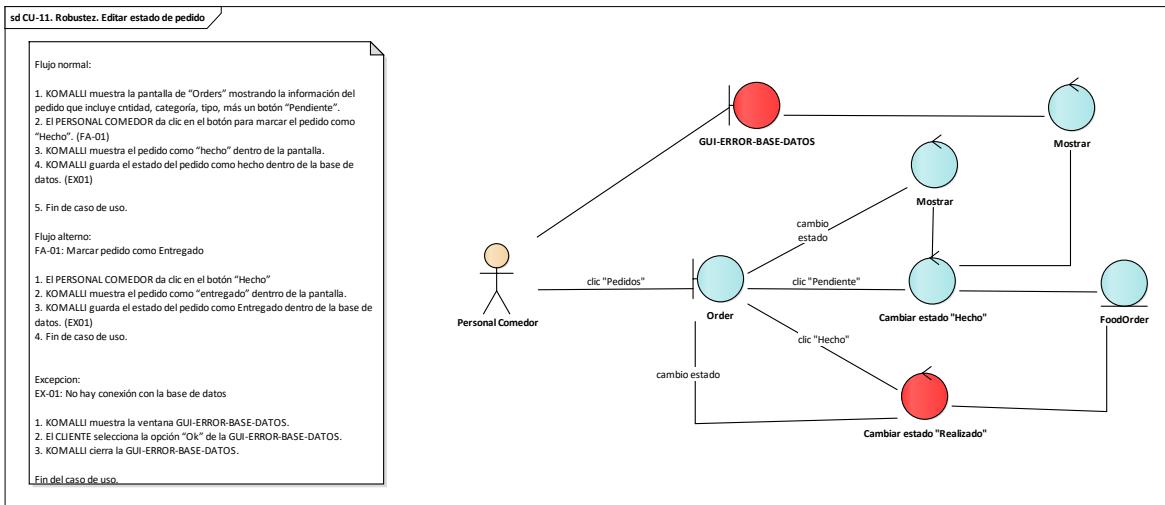


Ilustración 47 Diagrama de robustez CU-11

3.11.4 Diagrama de secuencia

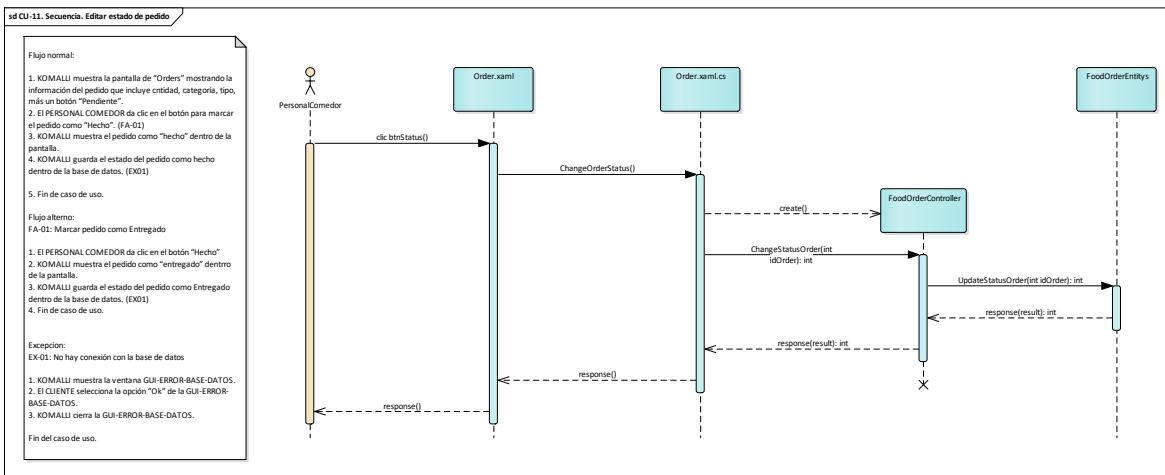


Ilustración 48 Diagrama de secuencia CU-11

3.12 CU 12 Publicar Menú

3.12.1 Prototipo

Este es un prototipo de la interfaz de usuario para publicar menús en la aplicación Komalli. La barra lateral izquierda muestra el logo Komalli, menús para Pedidos, Menú (con subopciones Publicar, Modificar, Eliminar), Bitacora y una sección de perfil con el nombre Miriam Ramírez Zárate. El cuadro principal tiene un título "Publicar Menú". Los campos obligatorios están marcados con asteriscos (*). Los campos incluyen Entrada, *Plato Fuerte, Guarnición, Ensalada, *Bebida, *Fecha (calendario) y *Tipo (desplegable). Una nota indica que las secciones con asterisco son obligatorias. Un botón "Publicar" finaliza el formulario.

Ilustración 49 Prototipos CU-12

Este es un prototipo de la interfaz de usuario para publicar menús en la aplicación Komalli, similar al anterior pero con datos llenados. Los campos Entrada, *Plato Fuerte, Guarnición, Ensalada, *Bebida y *Fecha están llenos con valores específicos. Una notificación emergente dice "Menú publicado con éxito". El calendario muestra que el día 4 está seleccionado. Los demás campos y la nota obligatoria permanecen igualmente.

Ilustración 50 Prototipos CU-12

Komalli

Pedidos
Menú
Publicar
Modificar
Eliminar
Bitácora

Miriam Ramírez Zárate

Publicar Menú

Entrada: Sopa de pasta *Fecha:

*Plato Fuerte: Milanesa de pollo

Guarnición: Arroz blanco

Ensalada: Zanahoria cruda

*Bebida: Agua de limón *Tipo: Comida ▼

No se pueden publicar 2 menús del mismo tipo para el mismo día

Las secciones marcadas con * son obligatorias para la publicación de un menú

Publicar

Lu	Ma	Mi	Ju	Vi	Sa	Do
1	2	3				
7	8	9	10			
4	15	16	17			
11	22	23	24			
8	29	30	31			

Ilustración 51 Prototipos CU-12

3.12.2 Descripción

ID:	CU-12
Nombre del CU:	Publicar Menú
Requisito funcional:	RF-16: Se debe permitir a los empleados planificar y organizar los menús semanales.
Descripción:	Se busca que el personal de cocina pueda publicar la información y contenidos de los menús semanales, para su correcta administración
Actor(es):	Personal de cocina (primario)
Disparador:	El PERSONAL DE COCINA selecciona la opción de publicar dentro de la sección de menú, del home correspondiente
Precondiciones:	<p>PRE-01.- El usuario debe estar registrado en el sistema</p> <p>PRE-02.- El usuario debe de tener su perfil habilitado dentro del sistema.</p> <p>PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Personal de cocina).</p>
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI muestra un formulario para el registro del menú semanal con los campos: <i>Entrada, plato fuerte guarnición, ensalada, bebida, tipo de comida, fecha</i>, siendo (<i>plato fuerte, bebida, tipo y fecha</i>) campos obligatorios para el proceso de publicación. 2. El PERSONAL DE COCINA llena los campos correspondientes y da clic en el botón de publicar. 3. KOMALLI verifica y valida los datos del Menú a registrar. (Ver FA-01) (Ver FA.02) 4. KOMALLI guarda la información del Menú en la base de datos y muestra una ventana de “<u>REGISTRO-MENÚ-EXITOSO</u>”. (Ver EX-01) 5. El PERSONAL DE COCINA selecciona la opción “aceptar” 6. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Publicación de más de 2 menús para un mismo día</p> <ol style="list-style-type: none"> 1. KOMALLI detecta que ya se cuentan con los 2 menús permitidos por día y muestra la alerta “<u>REGISTRO-MENÚ-INVALIDO</u>”. 2. El PERSONAL DE COCINA selecciona la opción “aceptar”. 3. Regresa al paso 2 del flujo normal. <p>FA-02: Ingreso de 2 menús del mismo tipo para el mismo día.</p> <ol style="list-style-type: none"> 1. KOMALLI detecta que ya se encuentra registrada con un menú del mismo tipo para el día que se desea registrar y muestra la alerta “<u>REGISTRO-MENÚ-INVALIDO</u>”. 2. El PERSONAL DE COCINA selecciona la opción “aceptar”. 3. Regresa al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El PERSONAL DE COCINA selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	<p>POST-01: Un nuevo menú es registrado en la Base de datos.</p> <p>POST-02: KOMALLI limpia los campos en caso de que el PERSONAL DE COCINA desee registrar otro menú.</p>
Incluye:	N/A
Extiende:	N/A

Tabla 12Descripción CU-12

3.12.3 Diagrama de robustez

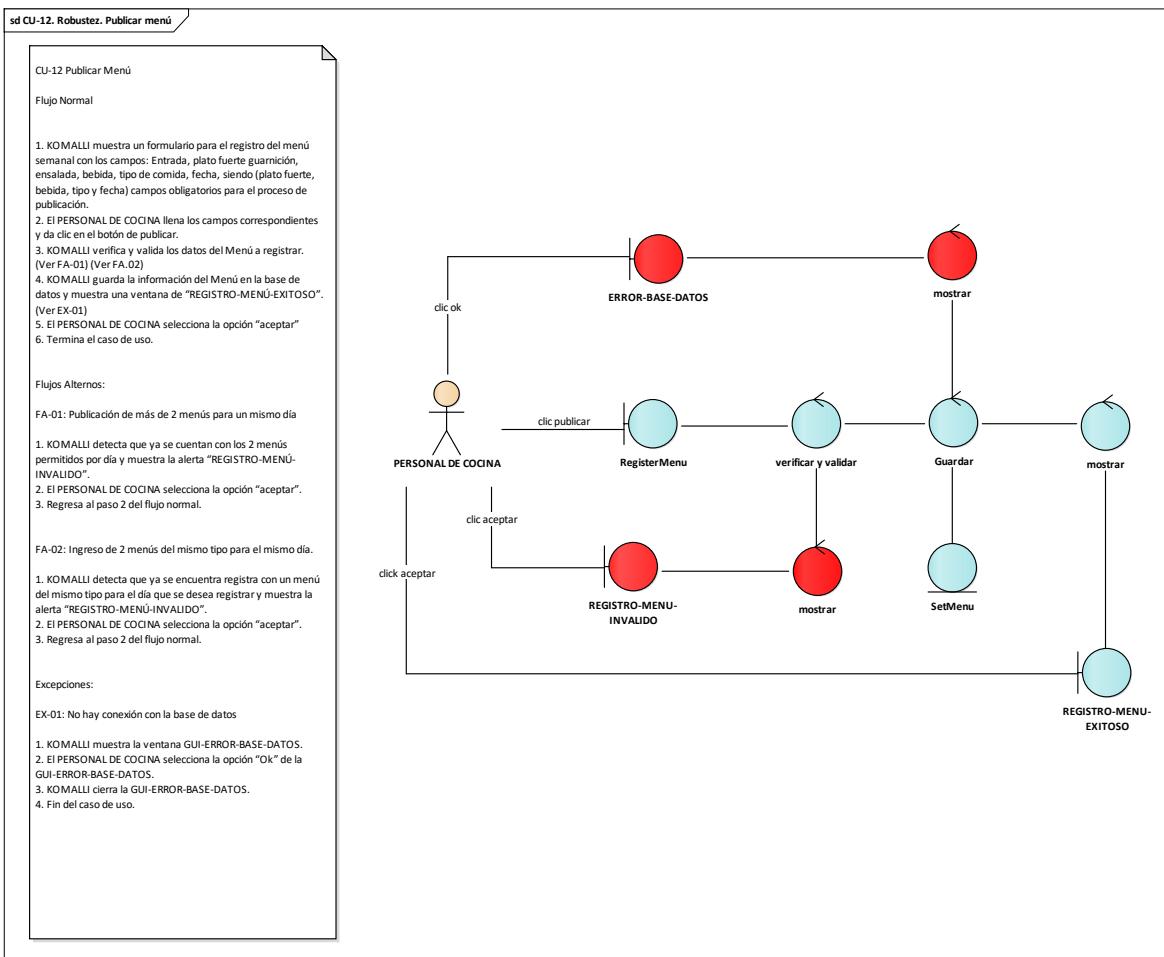


Ilustración 52 Diagrama de robustez CU-12

3.12.4 Diagrama de secuencia

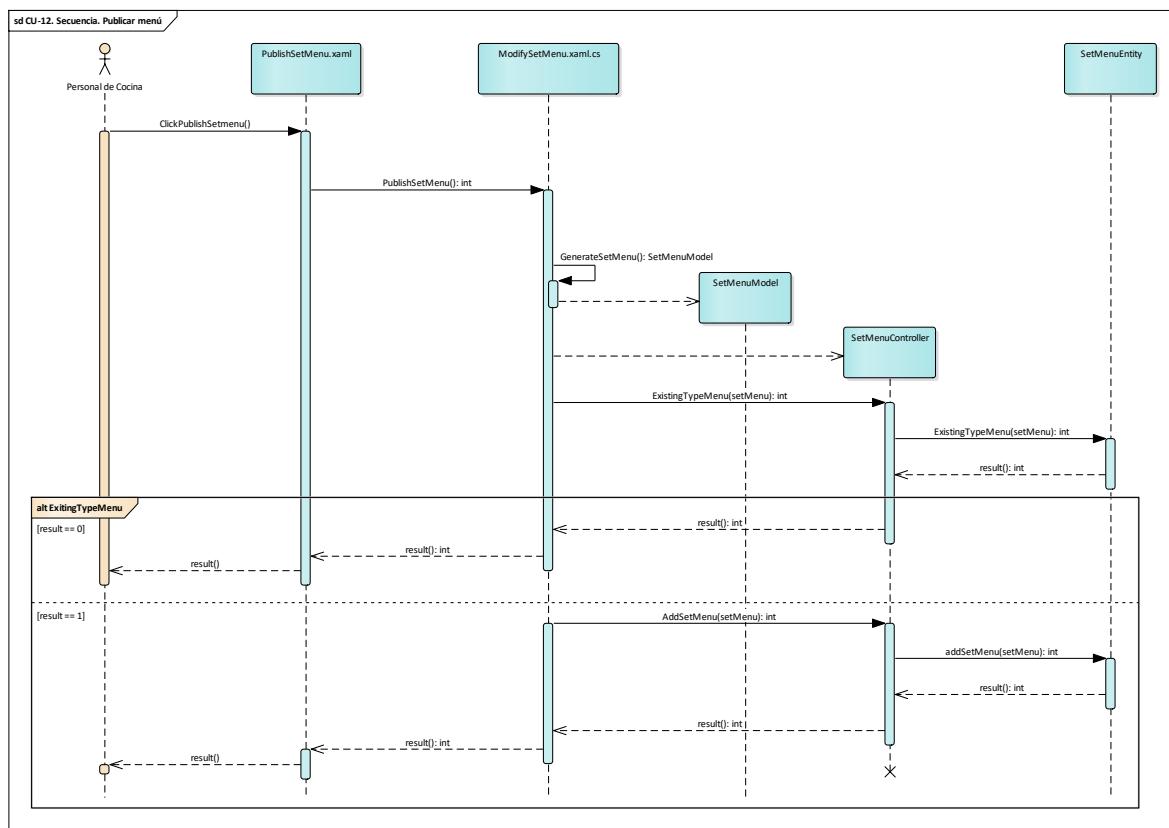


Ilustración 53 Diagrama de secuencia CU-12

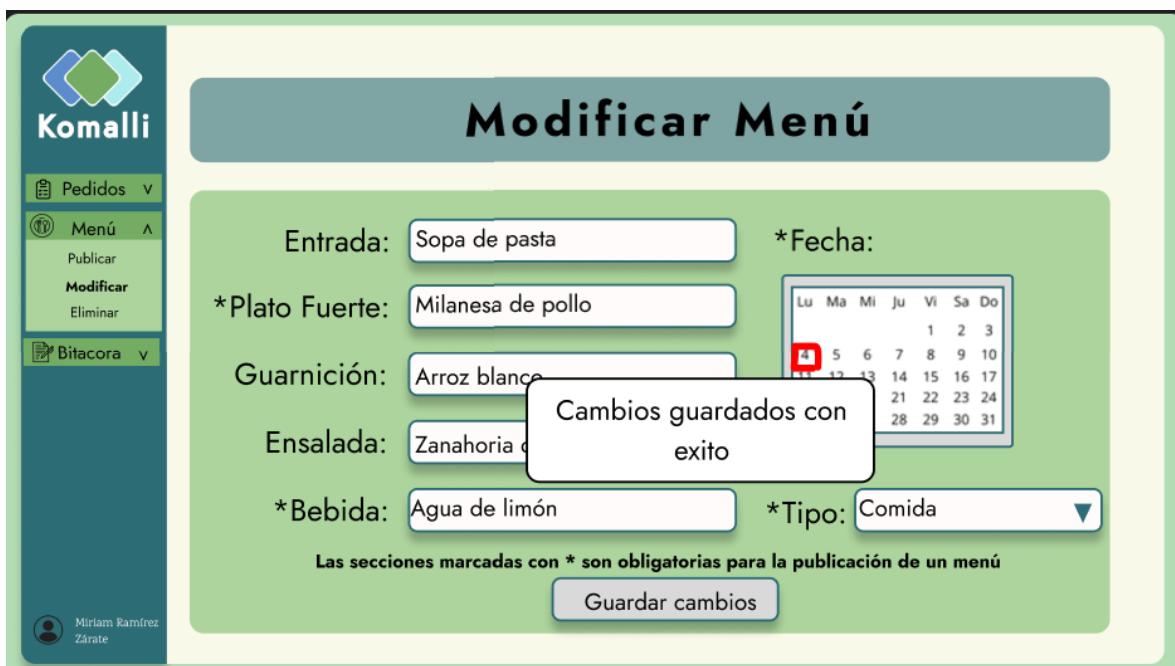
3.13 CU 13 Editar Menú

3.13.1 Prototipo



The screenshot shows the 'Modificar Menú' (Edit Menu) page of the Komalli application. The left sidebar has a 'Menú' section with 'Modificar' selected. The main area has a title 'Modificar Menú' and a subtitle 'Selecciona la fecha y el tipo de menú que deseas buscar para editar'. It includes a date picker labeled 'Fecha:' showing a grid from Monday to Sunday with days 1-31, and a dropdown menu labeled 'Tipo:' with a downward arrow. A 'Buscar menu' button is at the bottom.

Ilustración 54 Prototipo CU-13



This screenshot shows the 'Modificar Menú' page after changes have been made. The date '4' is highlighted in red in the calendar. A message box says 'Cambios guardados con éxito'. The date picker now shows a grid from Monday to Sunday with days 1-31. The dropdown menu 'Tipo:' now shows 'Comida'. The bottom message states 'Las secciones marcadas con * son obligatorias para la publicación de un menú'.

Ilustración 55 Prototipo CU-13

Komalli

Pedidos ▾

Menú ▾
Publicar
Modificar
Eliminar

Bitacora ▾

Miriam Ramírez Zárate

Modificar Menú

Entrada: Sopa de pasta *Fecha:

*Plato Fuerte: Milanesa de pollo

Guarnición: Arroz blanco

Ensalada: Zanahoria

*Bebida: Agua de limón *Tipo: Comida ▼

No se pueden publicar 2 menús del mismo tipo para el mismo día

Las secciones marcadas con * son obligatorias para la publicación de un menú

Guardar cambios

Lu	Ma	Mi	Ju	Vi	Sa	Do
1	2	3				
4	5	6	7	8	9	10
			14	15	16	17
			21	22	23	24
			28	29	30	31

Ilustración 56 Prototipo CU-13

3.13.2 Descripción

ID:	CU-13
Nombre del CU:	Editar Menú
Requisito funcional:	RF-16: Se debe permitir a los empleados planificar y organizar los menús semanales.
Descripción:	Se busca que el personal de cocina pueda modificar la información y contenidos de los menús semanales, para su correcta administración
Actor(es):	Personal de cocina (primario)
Disparador:	El PERSONAL DE COCINA selecciona la opción de modificar dentro de la sección de menú, del home correspondiente
Precondiciones:	<p>PRE-01.- El usuario debe estar registrado en el sistema</p> <p>PRE-02.- El usuario debe de tener su perfil habilitado dentro del sistema.</p> <p>PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Personal de cocina).</p>
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI muestra un formulario para la búsqueda del menú, a través de la <i>fecha del menú y su tipo (desayuno o comida)</i>. 2. El PERSONAL DE COCINA llena los campos solicitados y da clic en el botón de buscar menú. 3. KOMALLI busca el menú que coincide con los datos de búsqueda ingresados. (Ver EX-01). 4. KOMALLI muestra un formulario con la información del menú solicitado con los campos: <i>Entrada, plato fuerte guarnición, ensalada, bebida, tipo de comida, fecha</i>, siendo (<i>plato fuerte, bebida, tipo y fecha</i>) campos obligatorios para el proceso de modificación. (Ver FA.03) 5. El PERSONAL DE COCINA modifica los campos que considere necesarios y da clic en el botón de publicar. 6. KOMALLI verifica y valida los datos del Menú a editar. (Ver FA-01) (Ver FA.02) 7. KOMALLI actualiza la información del Menú en la base de datos y muestra una ventana de “<u>MODIFICACION-MENÚ-EXITOSO</u>”. (Ver EX-01). 8. El PERSONAL DE COCINA selecciona la opción “aceptar” 9. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Actualización de más de 2 menús para un mismo día</p> <ol style="list-style-type: none"> 1. KOMALLI detecta que ya se cuentan con los 2 menús permitidos por día y muestra la alerta “<u>MODIFICACION-MENÚ-INVALIDO</u>”. 2. El PERSONAL DE COCINA selecciona la opción “aceptar”. 3. Regresa al paso 2 del flujo normal. <p>FA-02: Ingreso de 2 menús del mismo tipo para el mismo día.</p> <ol style="list-style-type: none"> 1. KOMALLI detecta que ya se encuentra registrada con un menú del mismo tipo para el día que se desea registrar y muestra la alerta “<u>MODIFICACION-MENÚ-INVALIDO</u>”. 2. El PERSONAL DE COCINA selecciona la opción “aceptar”. 3. Regresa al paso 2 del flujo normal. <p>FA-03: Menú no existente:</p> <ol style="list-style-type: none"> 1. KOMALLI no encuentra un menú, que coincida con los criterios de búsqueda del PERSONAL DE COCINA y muestra la alerta “<u>MENU-INEXISTENTE</u>” 2. EL PERSONAL DE COCINA, da click en “aceptar” 3. Regresa al paso 1 del flujo normal
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El PERSONAL DE COCINA selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>.

	4. Fin del caso de uso.
Postcondiciones:	POST-01: La información del menú es actualizado en la Base de datos. POST-02: KOMALLI redirige al PERSONAL DE COCINA a la ventana de búsqueda de menús, en caso de que desee modificar algún otro menú.
Incluye:	N/A
Extiende:	N/A

Tabla 13 Descripción CU-13

3.13.3 Diagrama de robustez

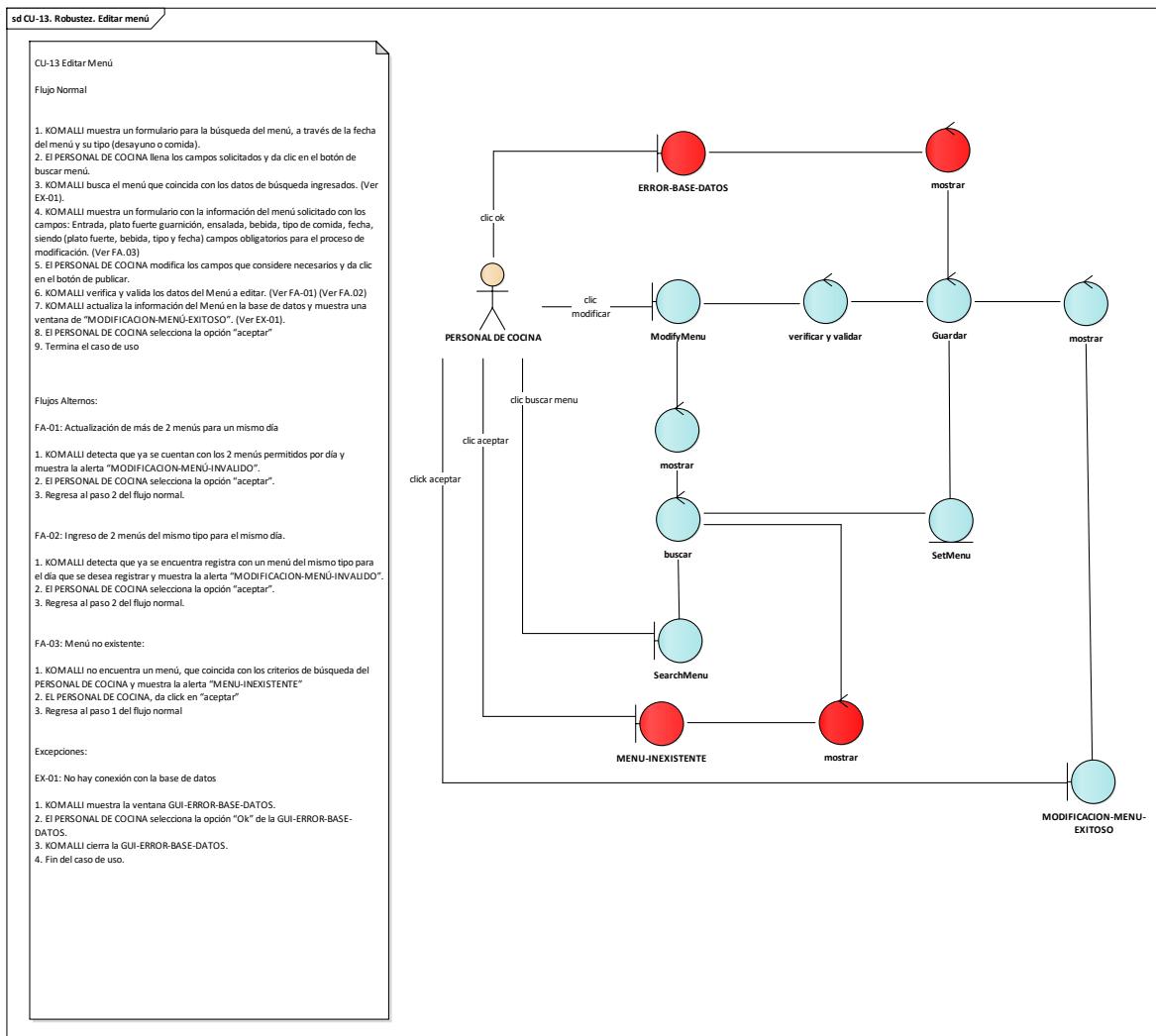


Ilustración 57 Diagrama de robustez CU-13

3.13.4 Diagrama de secuencia

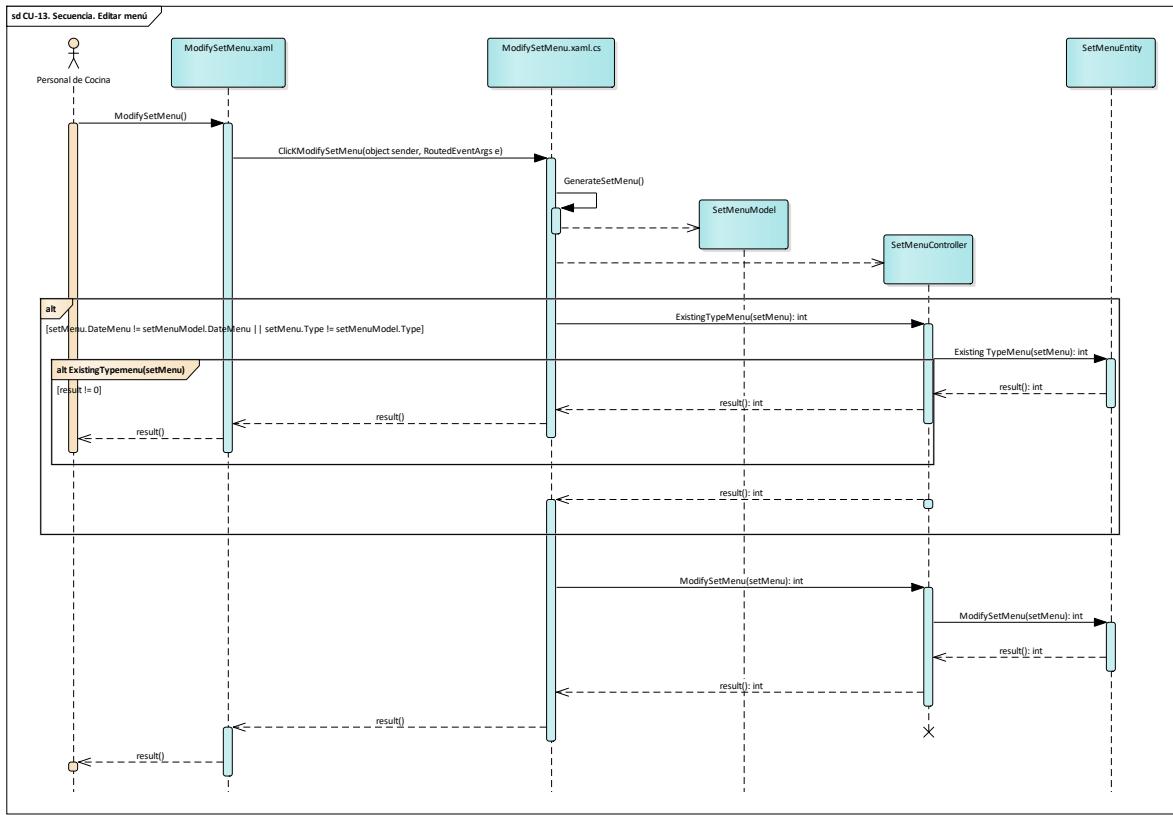


Ilustración 58 Diagrama de secuencia CU-13

3.14 CU 14 Eliminar Menú

3.14.1 Prototipo



The screenshot shows a user interface for deleting a menu. On the left, there's a vertical sidebar with the Komalli logo at the top, followed by 'Pedidos', 'Menú' (with 'Publicar', 'Modificar', 'Eliminar' options), and 'Bitácora'. Below the sidebar is a user profile for 'Miriam Ramírez Zárate'. The main area has a title 'Eliminar Menú' and a subtitle 'Selecciona la fecha y el tipo de menú que deseas buscar para eliminar'. It features a 'Fecha:' section with a calendar grid showing months from April to June. To the right of the calendar is a dropdown menu labeled 'Tipo:' with a downward arrow. At the bottom is a blue button labeled 'Buscar menú'.

Ilustración 59 Prototipo CU-14



This screenshot shows the same interface as above, but with additional validation. The 'Entrada:' field contains 'Sopa de pasta', the 'Plato Fuerte:' field contains 'Milanesa de pollo', and the 'Bebida:' field contains 'Agua de limón'. The 'Fecha:' field is highlighted with a red border. A modal dialog box is centered over the form, asking 'Esta seguro que quiere eliminar el menú?' with 'Si' and 'No' buttons. To the right of the modal, the 'Tipo:' dropdown is set to 'Comida'. At the bottom, a note says 'Las secciones marcadas con * son obligatorias para la publicación de un menú' and a blue 'Eliminar Menú' button is visible.

Ilustración 60 Prototipo CU-14

Ilustración 61 Prototipo CU-14

3.14.2 Descripción

ID:	CU-14
Nombre del CU:	Eliminar Menú
Requisito funcional:	RF-16: Se debe permitir a los empleados planificar y organizar los menús semanales.
Descripción:	Se busca que el personal de cocina pueda eliminar menús semanales, para su correcta administración
Actor(es):	Personal de cocina (primario)
Disparador:	El PERSONAL DE COCINA selecciona la opción de eliminar dentro de la sección de menú, del home correspondiente
Precondiciones:	PRE-01.- El usuario debe estar registrado en el sistema PRE-02.- El usuario debe tener su perfil habilitado dentro del sistema. PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Personal de cocina).
Flujo Normal	<ol style="list-style-type: none"> KOMALLI muestra un formulario para la búsqueda del menú, a través de la <i>fecha del menú y su tipo (desayuno o comida)</i>. El PERSONAL DE COCINA llena los campos solicitados y da clic en el botón de buscar menú. KOMALLI busca el menú que coincide con los datos de búsqueda ingresados. (Ver EX-01). KOMALLI muestra un formulario con la información del menú solicitado con los campos: <i>Entrada, plato fuerte guarnición, ensalada, bebida, tipo de comida, fecha</i>, siendo (<i>plato fuerte, bebida, tipo y fecha</i>). (Ver FA.01) El PERSONAL DE COCINA da clic en el botón “eliminar menú”. KOMALLI verifica que el PERSONAL DE COCINA realmente quiera eliminar el menú y muestra la alerta “<u>CONFIRMACION-ELIMINAR-MENU</u>”. El PERSONAL DE COCINA da clic en el botón “sí”. (ver. FA-02).

	<p>8. KOMALLI elimina la información del Menú en la base de datos y muestra una ventana de “<u>ELIMINACION-MENÚ-EXITOSO</u>”. (Ver EX-01).</p> <p>9. El PERSONAL DE COCINA selecciona la opción “aceptar”</p> <p>10. Termina el caso de uso.</p>
Flujos Alternos:	<p>FA-01: Menú no existente:</p> <ol style="list-style-type: none"> 1. KOMALLI no encuentra un menú, que coincida con los criterios de búsqueda del PERSONAL DE COCINA y muestra la alerta “<u>MENU-INEXISTENTE</u>” 2. EL PERSONAL DE COCINA, da click en “aceptar” 3. Regresa al paso 1 del flujo normal <p>FA-02: Cancelar eliminar menu:</p> <ol style="list-style-type: none"> 1. EL PERSONAL DE COCINA da clic en el botón “no” 2. Regresa al paso 4 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El PERSONAL DE COCINA selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	<p>POST-01: La información del menú es eliminada en la Base de datos.</p> <p>POST-02: KOMALLI redirige al PERSONAL DE COCINA a la ventana de búsqueda de menús, en caso de que desee eliminar algún otro menú.</p>
Incluye:	N/A
Extiende:	N/A

Tabla 14 14 Descripción CU-14

3.14.3 Diagrama de robustez

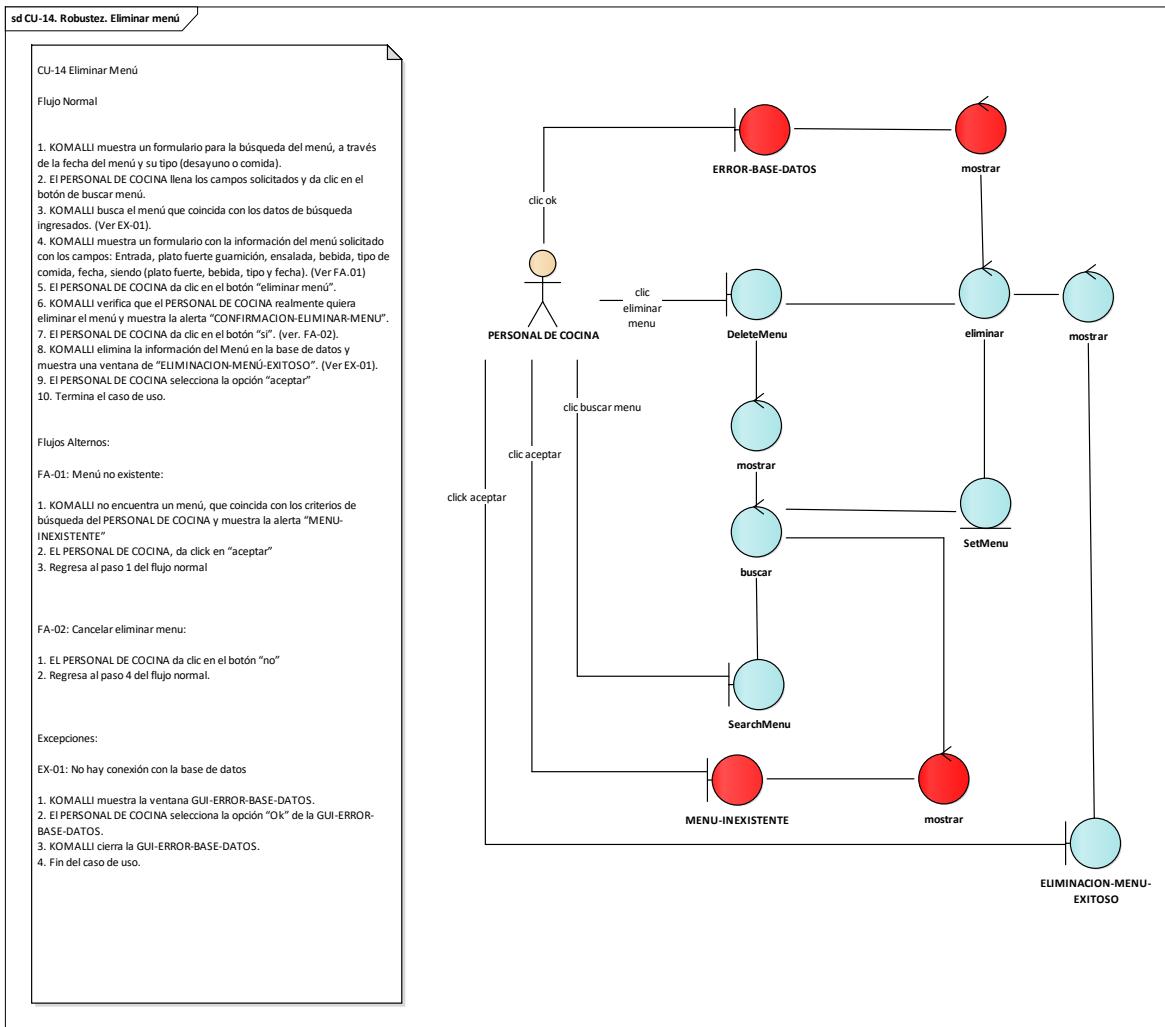


Ilustración 62 Diagrama de robustez CU-14

3.14.4 Diagrama de secuencia

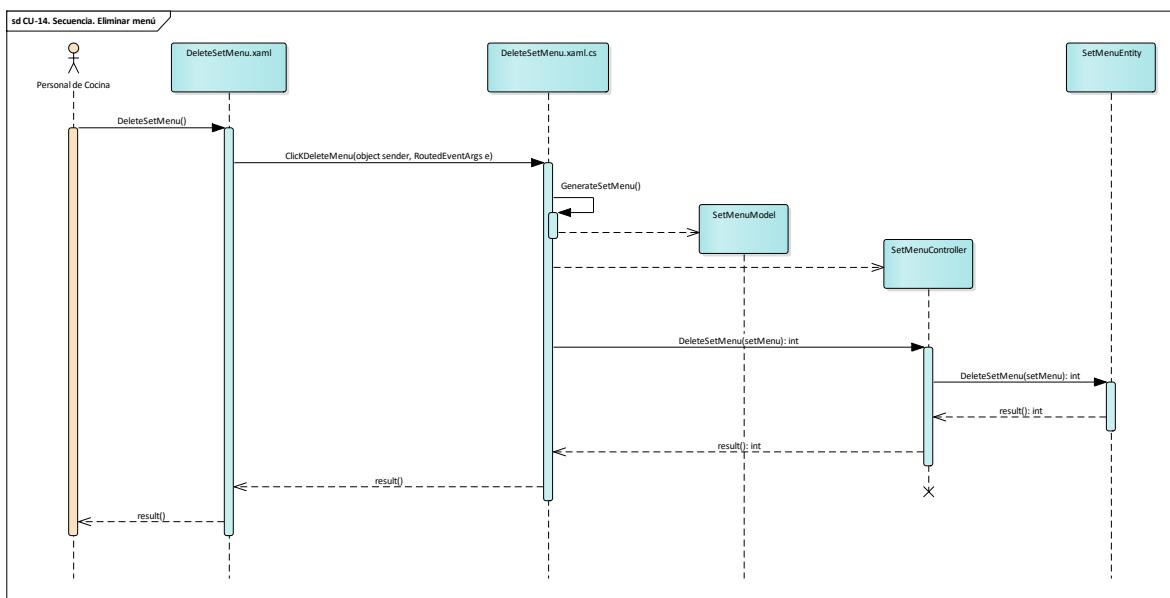


Ilustración 63 Diagrama de secuencia CU-14

3.15 CU 15 Consultar Menu

3.15.1 Prototipo



Ilustración 64 Prototipo CU-15



Ilustración 65 Prototipo CU-15

3.15.2 Descripción

ID:	CU-15
Nombre del CU:	Consultar Menú
Requisito funcional:	RF-13: El sistema debe permitir consultar los pedidos realizados en el kiosko y proceder a su cobro.
Descripción:	Se desea saber los menús existentes en el sistema, así como los distintos alimentos que hay dada distintas categorías.
Actor(es):	Cliente (Primario)
Disparador:	El cliente accede a la ventana del Kiosko.
Precondiciones:	PRE-01: Se ha registrado en el sistema por lo menos algún menú
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI recupera de la base de datos los platillos que fueron previamente registrados y los separa en distintas categorías mostrando el <i>nombre, precio e imagen</i> del alimento. (ver EX-01) 2. El CLIENTE selecciona la opción “Agregar” de algún alimento de alguna de las categorías. (ver FA-01) (ver FA-02) 3. KOMALLI agrega el alimento a la lista de los pedidos aumentando el subtotal de la cuenta del pedido. 4. El CLIENTE selecciona la opción “Continuar”. 5. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Agregar menú del día tipo estudiante</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana emergente <u>GUI-TIPO-ROL</u> preguntando al cliente el rol con el que cuenta. 2. El CLIENTE selecciona la opción “Sí” confirmando que se trata de un estudiante. 3. El flujo continua en el paso 2. <p>FA-02: Agregar menú del día tipo externo</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana emergente <u>GUI-TIPO-ROL</u> preguntando al cliente el rol con el que cuenta. 2. El CLIENTE selecciona la opción “No” confirmando que no es un estudiante. 3. El flujo continua en el paso 2.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El CLIENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	<p>POST-01: Se agrega un nuevo alimento a la lista de pedidos.</p> <p>POST-02: Se aumenta el subtotal a la cuenta de los pedidos del cliente.</p>
Incluye:	N/A
Extiende:	CU-09, CU-14, CU-25

Tabla 1515 Descripción CU-15

3.15.3 Diagrama de robustez

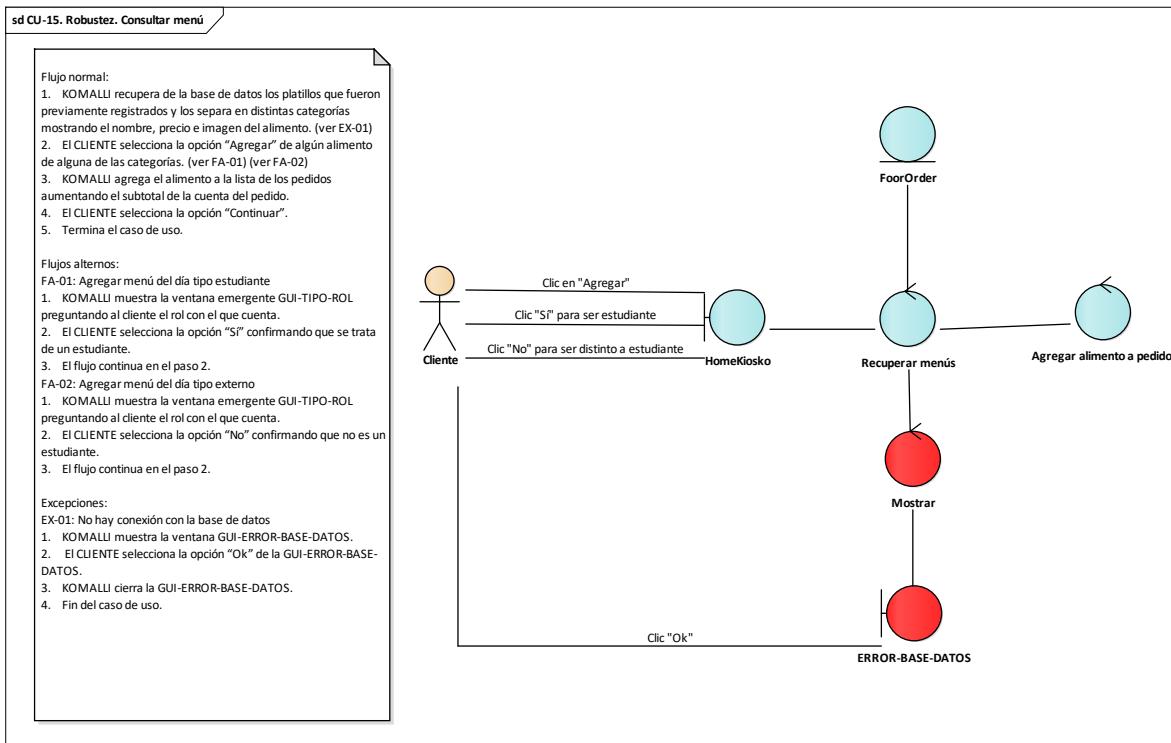


Ilustración 66 Diagrama robustez CU-15

3.15.4 Diagrama de secuencia

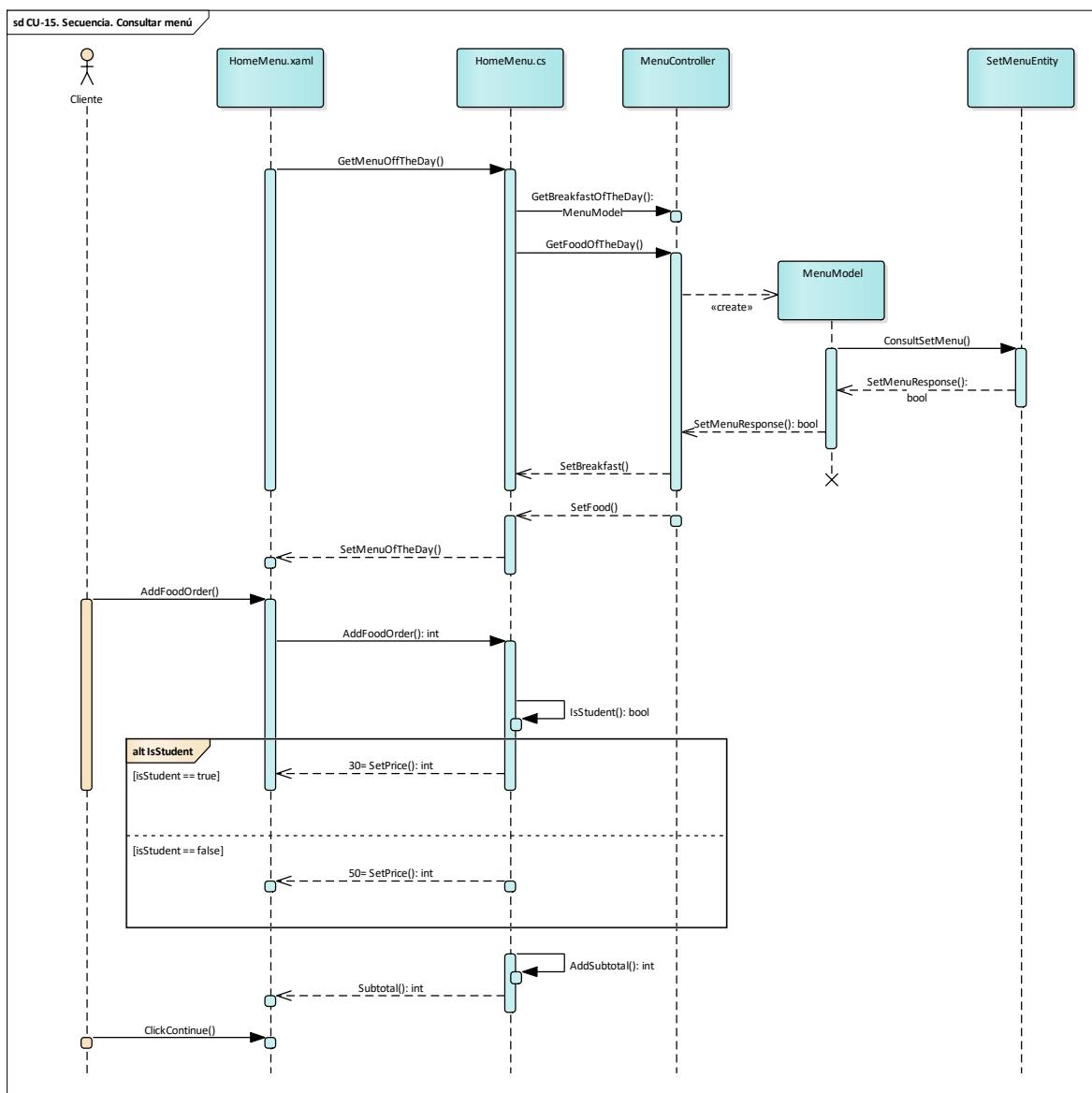


Ilustración 67 Diagrama secuencia CU-15

3.16 CU 16 Agregar Comentario a bitácora

3.16.1 Prototipo



Ilustración 68 Prototipo CU-16

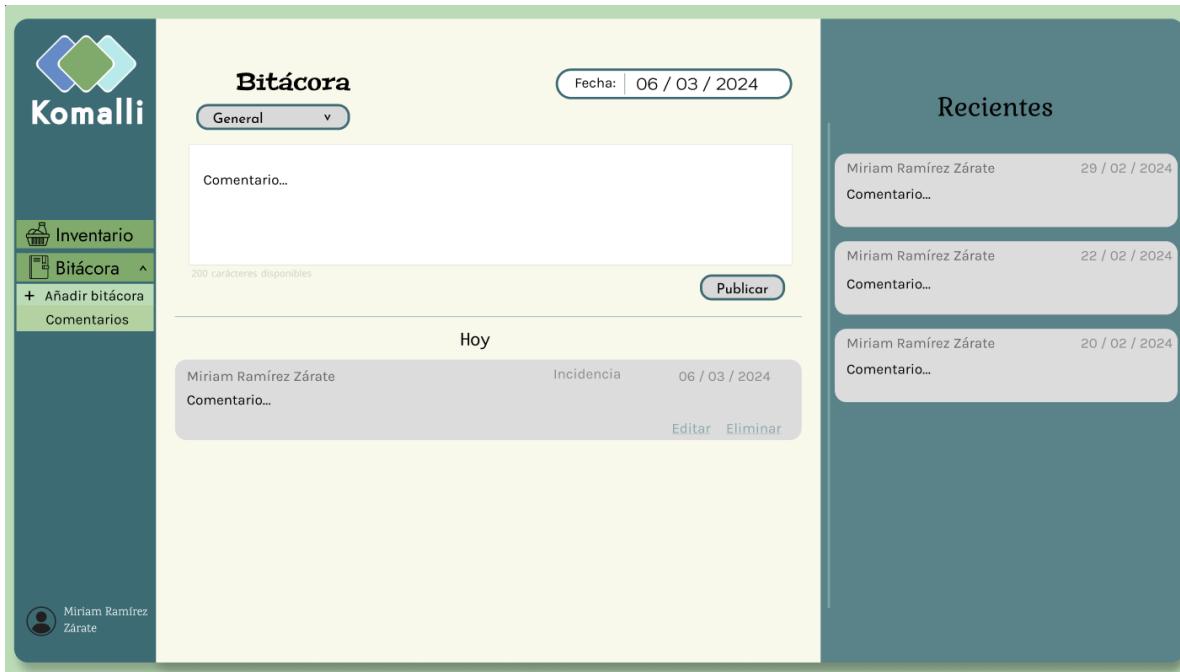


Ilustración 69 Prototipo CU-16

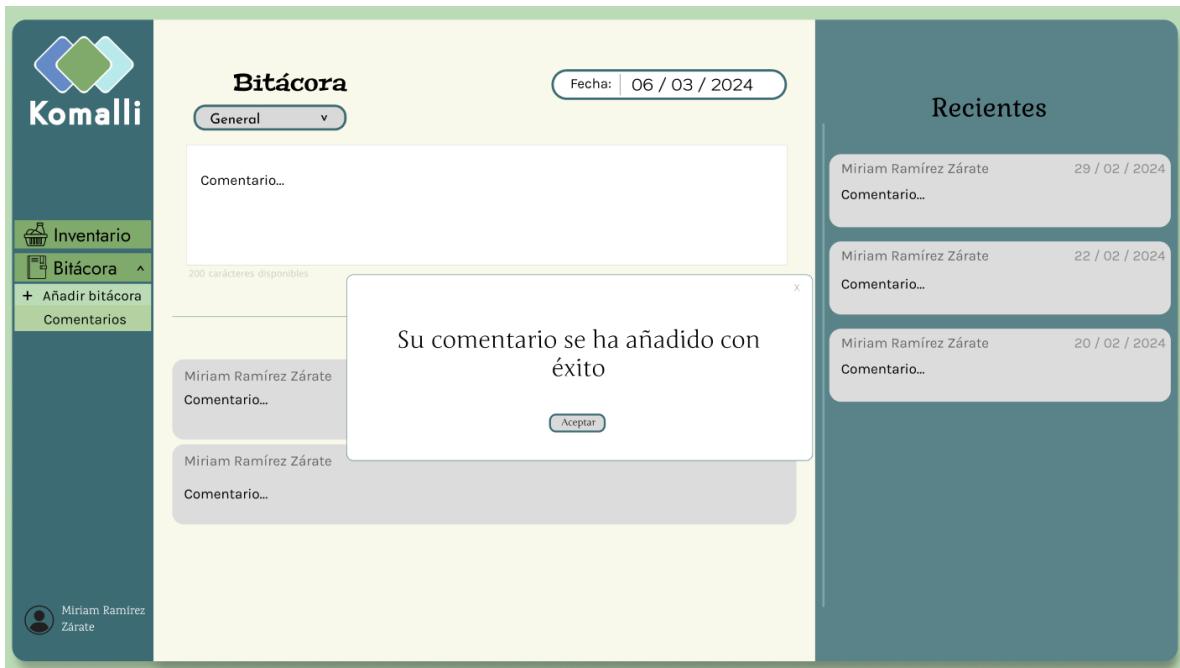


Ilustración 70 Prototipo CU-16

3.16.2 Descripción

ID:	CU-16
Nombre del CU:	Agregar comentario a bitácora
Requisito funcional:	RF-16: Komalli permite administrar por medio de una bitácora aquellos sucesos o imprevistos que puedan suscitarse a lo largo de la jornada laboral dentro del comedor.
Descripción:	Permite el Jefe de Cocina ingresar al sistema observaciones o datos asociados a las actividades dentro del comedor universitario.
Actor(es):	Jefe de Cocina (Primario)
Disparador:	El Jefe de Cocina da clic en Bitácora dentro de la barra de menú.
Precondiciones:	PRE-01: El Jefe de Cocina debe tener un perfil activo dentro del sistema
Flujo Normal:	<p>1. KOMALLI muestra dentro de la pantalla “Logbook” un <i>cuadro de texto</i> para agregar comentario, un botón Publicar, la <i>fecha del día en que se accede</i> a la Bitácora, los comentarios recientes y los del día realizados por el Jefe de Cocina y una lista de los <i>comentarios añadidos</i> que contiene: <i>autor del comentario, fecha en la cual se ha añadido el comentario, el contenido</i>.</p> <p>2. El JEFE DE COCINA escribe dentro del <i>cuadro de texto</i>, dando clic en el botón “Publicar”.</p> <p>3. KOMALLI valida el contenido ingresado dentro del <i>cuadro de texto</i>. (FA-01)</p> <p>4. KOMALLI guarda la información ingresada dentro del <i>cuadro de texto</i> mostrando un mensaje de registro exitoso. (EX01)</p> <p>5. El JEFE DE COCINA da clic en el botón “Cerrar” de la ventana emergente.</p> <p>6. KOMALLI cierra la ventana emergente.</p> <p>7. Fin de caso de uso.</p>
Flujos Alternos:	<p>FA-01: Excedente de caracteres permitidos</p> <p>1. KOMALLI muestra una ventana emergente con un mensaje donde se indica que se ha excedido el límite de caracteres.</p> <p>2. El JEFE DE COCINA da clic en el botón “Cerrar” de la ventana emergente.</p> <p>3. KOMALLI cierra la ventana emergente.</p> <p>4. Fin de caso de uso.</p>
Excepciones:	<p>EX01: Sin conexión a base de datos</p> <p>1. KOMALLI muestra una ventana emergente con un mensaje sobre el error de conexión a la base de datos.</p> <p>2. El JEFE DE COCINA da clic en el botón de “Cerrar” de la ventana emergente.</p> <p>3. KOMALLI cierra la ventana emergente.</p> <p>4. Fin de caso de uso.</p>
Postcondiciones:	<p>POST-01: Komalli ha agregado un nuevo comentario dentro de Bitácora dentro de la base de datos.</p> <p>POST-02: Komalli muestra el comentario añadido dentro de la lista de comentarios realizados dentro de Bitácora.</p>
Incluye:	N/A
Extiende:	N/A

Tabla 16. Descripción CU-16

3.16.3 Diagrama de robustez

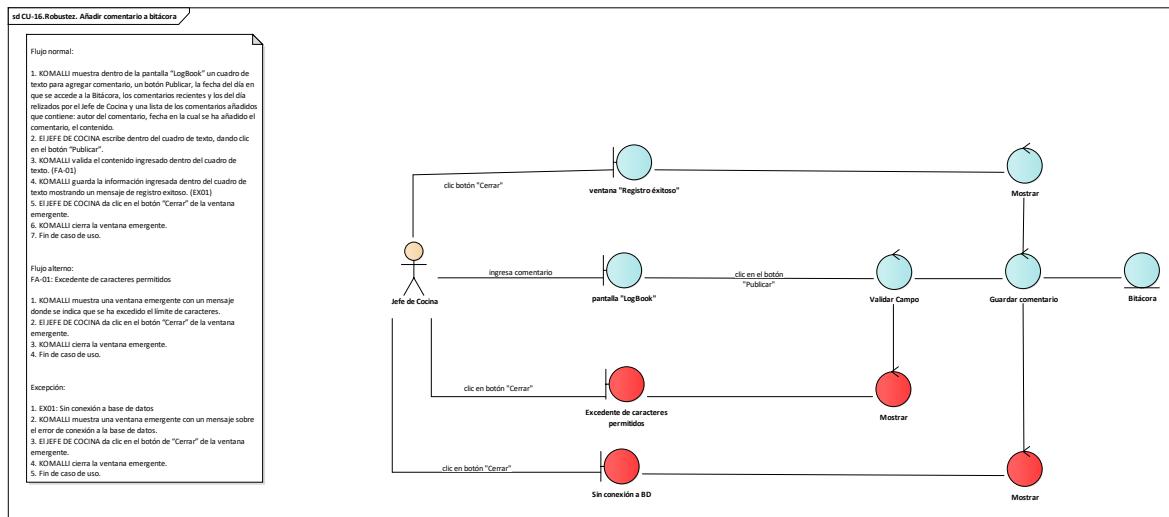


Ilustración 71 Diagrama de robustez CU-16

3.16.4 Diagrama de secuencia

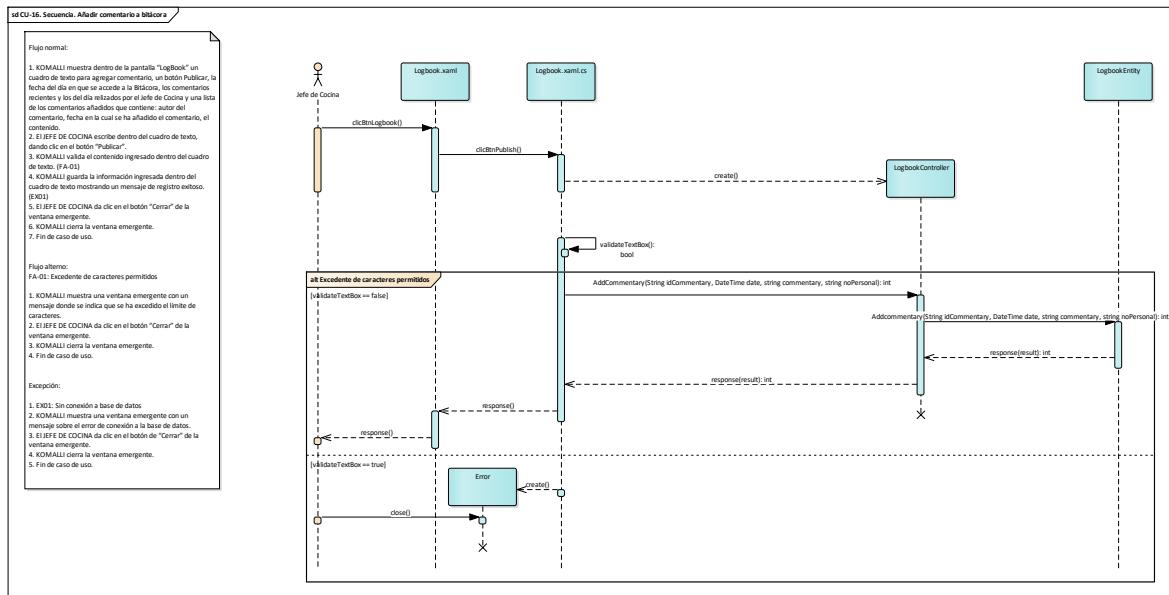


Ilustración 72 Diagrama de secuencia CU-16

3.17 CU 17 Editar comentario de bitácora

3.17.1 Prototipo

El prototipo muestra la interfaz de usuario para editar un comentario en la sección de Bitácora. En la parte superior izquierda, el logo de Komalli y el menú lateral incluyen las opciones: Inventario, Bitácora (seleccionada), Añadir bitácora y Mis comentarios. El nombre del usuario, Miriam Ramírez Zárate, aparece en el pie de página.

Bitácora

General

Fecha: 06 / 03 / 2024

Comentario...

200 caracteres disponibles

Cancelar Modificar

Comentarios añadidos

Miriam Ramírez Zárate	Incidencia	06 / 03 / 2024	Eliminar
Comentario...			

Miriam Ramírez Zárate	General	05 / 03 / 2024	Eliminar
Comentario...			

Recientes

Miriam Ramírez Zárate	29 / 02 / 2024
Comentario...	

Miriam Ramírez Zárate	22 / 02 / 2024
Comentario...	

Miriam Ramírez Zárate	20 / 02 / 2024
Comentario...	

Ilustración 73 Prototipo CU-17



Ilustración 74 Prototipo CU-17

3.17.2 Descripción

ID:	CU-17
Nombre del CU:	Editar comentario de bitácora
Requisito funcional:	RF-16: Komalli permite administrar por medio de una bitácora aquellos sucesos o imprevistos que puedan suscitarse a lo largo de la jornada laboral dentro del comedor.
Descripción:	Permite al Jefe de cocina modificar las observaciones realizadas en Bitácora durante el día
Actor(es):	Jefe de Cocina (Primario)
Disparador:	El Jefe de Cocina da clic en “Editar” del comentario que desea realizar una modificación
Precondiciones:	PRE-01: Solo se pueden editar los comentarios durante el mismo día de haberse añadido PRE-02: Solo se pueden editar comentarios si pertenecientes a la cuenta de usuario que lo registró
Flujo Normal:	<ol style="list-style-type: none"> KOMALLI muestra en la pantalla “EditLogBook” un Textbox con el comentario que se quiere modificar. El JEFE DE COCINA realiza los cambios deseados dentro del comentario realizado. El JEFE DE COCINA da clic en el botón “Modificar” (FA-02) KOMALLI muestra una ventana emergente con un mensaje preguntando si se desea realizar dicho cambio, junto a dos botones “Cancelar” y “Aceptar”. (FA-01) El JEFE DE COCINA da clic en “Aceptar”. KOMALLI valida el contenido ingresado dentro del cuadro de texto. (FA-03) KOMALLI guarda el contenido del comentario actualizado dentro de la base de datos. (EX01) Fin de caso de uso

Flujos Alternos:	<p>FA-01: Cancelar comentario modificado</p> <ol style="list-style-type: none"> 1. El JEFE DE COCINA da clic en “Cancelar” dentro de la ventana emergente de confirmación de actualización de comentario. 2. KOMALLI cierra la ventana emergente, regresando al flujo normal número 1. <p>FA-02: Cancelar modificación</p> <ol style="list-style-type: none"> 1. El JEFE DE COCINA da clic en el botón “Cancelar” dentro de la ventana “EditLogBook ” 2. KOMALLI regresa a la pantalla “ LogBook ” 3. Fin de caso de uso. <p>FA-03: Excedente de caracteres permitidos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra una ventana emergente con un mensaje donde se indica que se ha excedido el límite de caracteres. 2. El JEFE DE COCINA da clic en el botón “Cerrar” de la ventana emergente. 3. KOMALLI cierra la ventana emergente. 4. Fin de caso de uso.
Excepciones:	<p>EX01: Sin conexión a la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra una Ventana emergente con un mensaje sobre el error de conexión a la base de datos. 2. El JEFE DE COCINA da clic en el botón “Cerrar” de la ventana emergente. 3. KOMALLI cierra la ventana emergente. 4. Fin de caso de uso.
Postcondiciones:	<p>POST-01: Komalli actualiza el comentario dentro de la base de datos</p> <p>POST-02: Komalli muestra la actualización del comentario en pantalla</p>
Incluye:	N/A
Extiende:	N/A

Tabla 17. Descripción CU-17

3.17.3 Diagrama de robustez

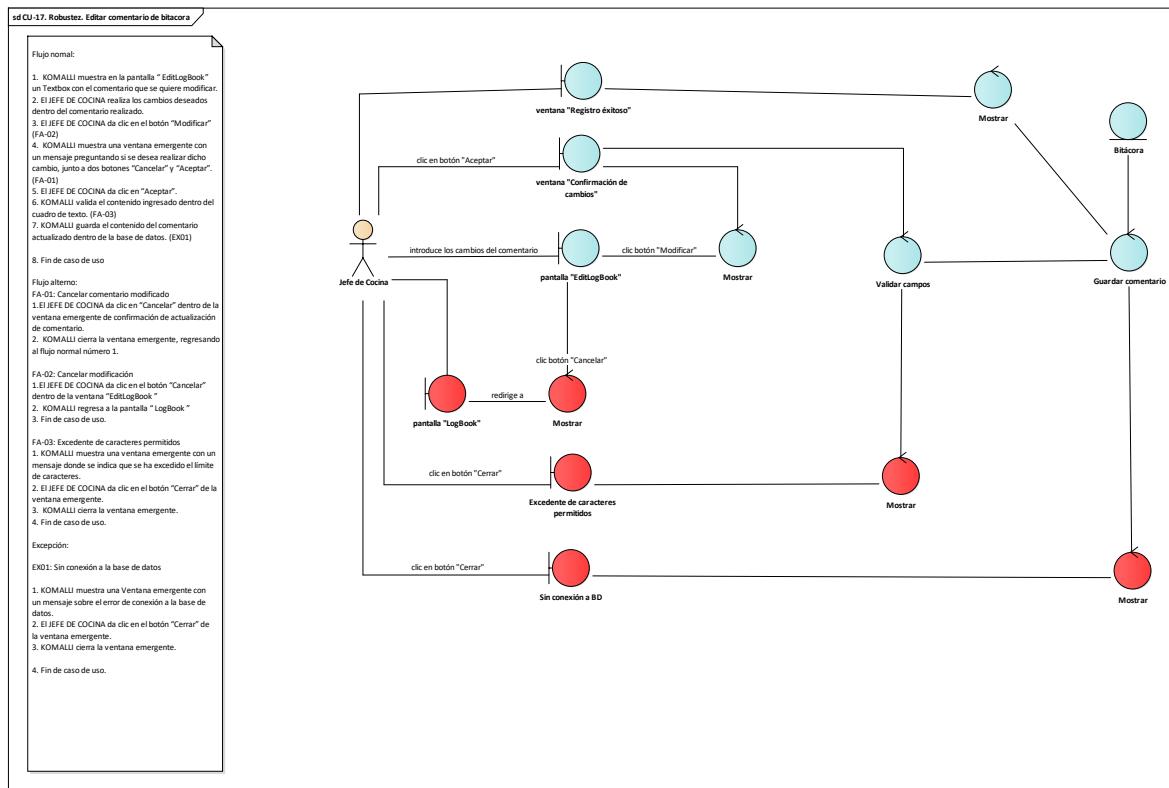


Ilustración 75 Diagrama robustez CU-17

3.17.4 Diagrama de secuencia

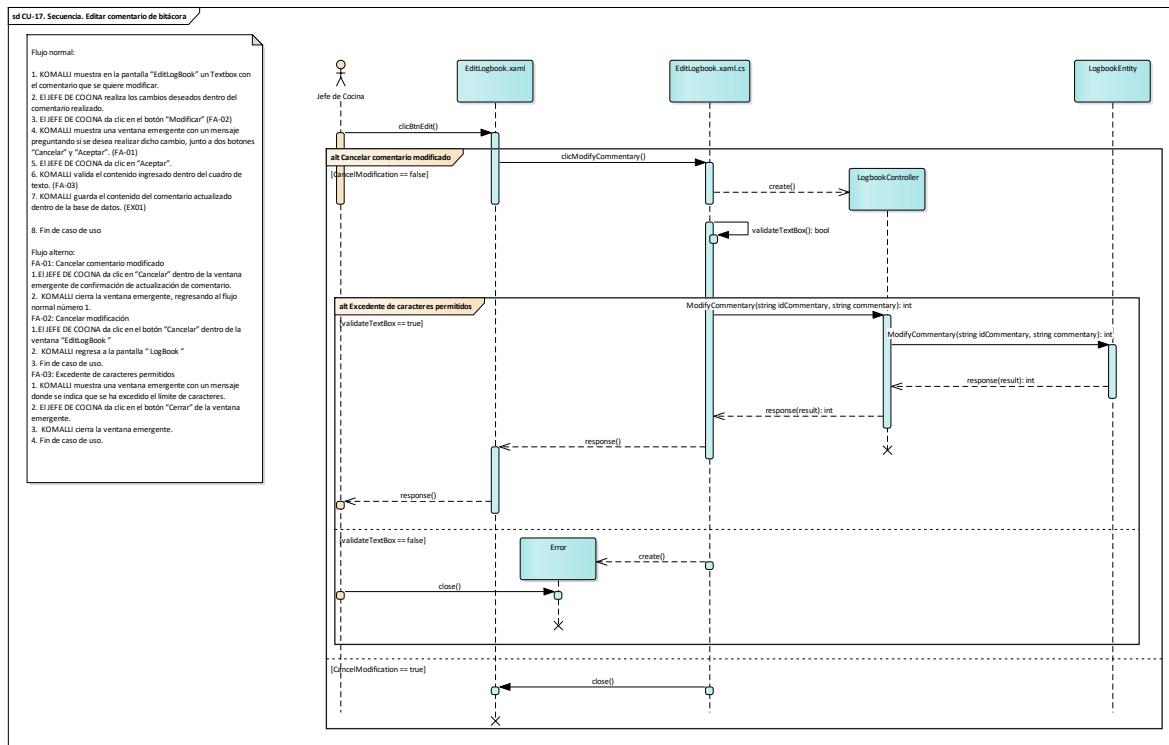


Ilustración 76 Diagrama secuencia CU-17

3.18 CU 18 Eliminar comentario de bitácora

3.18.1 Prototipo

El prototipo muestra la interfaz de usuario para la administración de la bitácora. En la parte superior, se encuentra el logo de Komalli y la fecha actual (06 / 03 / 2024). La barra lateral izquierda incluye menús para 'Inventario', 'Bitácora' (seleccionado), 'Añadir bitácora' y 'Comentarios'. El contenido principal muestra una lista de entradas en la bitácora con información detallada y opciones para editar o eliminar cada uno. Una sección lateral derecha titulada 'Recientes' enumera los últimos tres comentarios realizados por Miriam Ramírez Zárate.

Nombre	Tipo	Fecha	Opciones
Miriam Ramírez Zárate	Incidencia	06 / 03 / 2024	Editar Eliminar
Miriam Ramírez Zárate	Merma	06 / 03 / 2024	Editar Eliminar
Miriam Ramírez Zárate	Comentario...	29 / 02 / 2024	Comentario...
Miriam Ramírez Zárate	Comentario...	22 / 02 / 2024	Comentario...
Miriam Ramírez Zárate	Comentario...	20 / 02 / 2024	Comentario...

Ilustración 77 Prototipo CU-18

Este cuadro detallado muestra dos entradas de la bitácora:

Miriam Ramírez Zárate	Editar	Incidencia	06 / 03 / 2024
Comentario...			Editar Eliminar
Miriam Ramírez Zárate	Editar	Merma	06 / 03 / 2024
Comentario...			Editar Eliminar

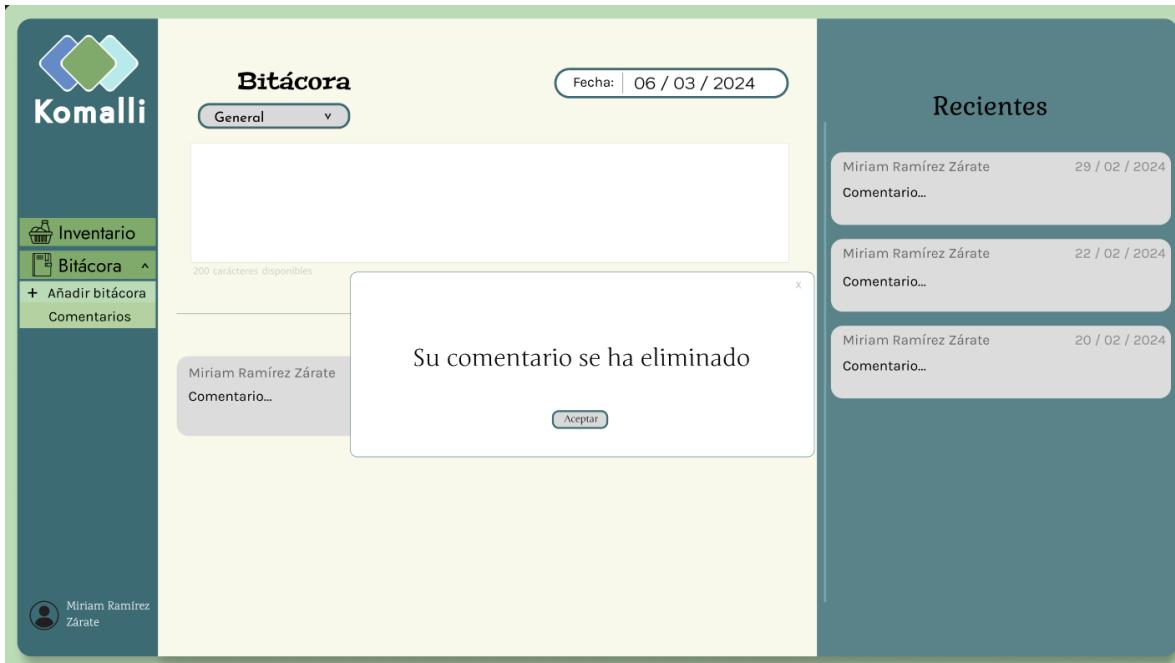


Ilustración 78 Prototipo CU-18

3.18.2 Descripción

ID:	CU-18
Nombre del CU:	Eliminar comentario de bitácora
Requisito funcional:	RF-16: Komalli permite administrar por medio de una bitácora aquellos sucesos o imprevistos que puedan suscitarse a lo largo de la jornada laboral dentro del comedor.
Descripción:	Permite al Jefe de cocina eliminar las observaciones realizadas en Bitácora durante el día
Actor(es):	Jefe de Cocina (Primario)
Disparador:	El Jefe de Cocina da clic en “Editar” del comentario que desea realizar una modificación
Precondiciones:	PRE-01: Solo se pueden eliminar los comentarios durante el mismo día de haberse añadido PRE-02: Solo se pueden eliminar comentarios si pertenecientes a la cuenta de usuario que lo registró
Flujo Normal:	<ol style="list-style-type: none"> KOMALLI muestra en la pantalla “EditLogBook” un Textbox con el comentario que se quiere eliminar. El JEFE DE COCINA da clic dentro del ícono de basura. KOMALLI muestra una ventana emergente con un mensaje preguntando si se desea eliminar dicho comentario, junto a dos botones “Cancelar” y “Aceptar”. (FA-01) El JEFE DE COCINA da clic en “Aceptar”. KOMALLI muestra un mensaje sobre que el comentario se ha eliminado con éxito. KOMALLI elimina el contenido del comentario dentro de la base de datos. (EX01) Fin de caso de uso
Flujos Alternos:	<p>FA-01: Cancelar eliminación</p> <ol style="list-style-type: none"> El JEFE DE COCINA da clic en “Cancelar” dentro de la ventana emergente de confirmación de eliminación de comentario. KOMALLI cierra la ventana emergente, regresando al flujo normal número 1.

Excepciones:	EX01: Sin conexión a la base de datos 1. KOMALLI muestra una Ventana emergente con un mensaje sobre el error de conexión a la base de datos. 2. EL JEFE DE COCINA da clic en el botón “Cerrar” de la ventana emergente. 3. KOMALLI cierra la ventana emergente. 4. Fin de caso de uso.
Postcondiciones:	POST-01: Komalli elimina el comentario dentro de la base de datos POST-02: Komalli no muestra el comentario en pantalla
Incluye:	N/A
Extiende:	N/A

Tabla 18. Descripción CU-18

3.18.3 Diagrama de robustez

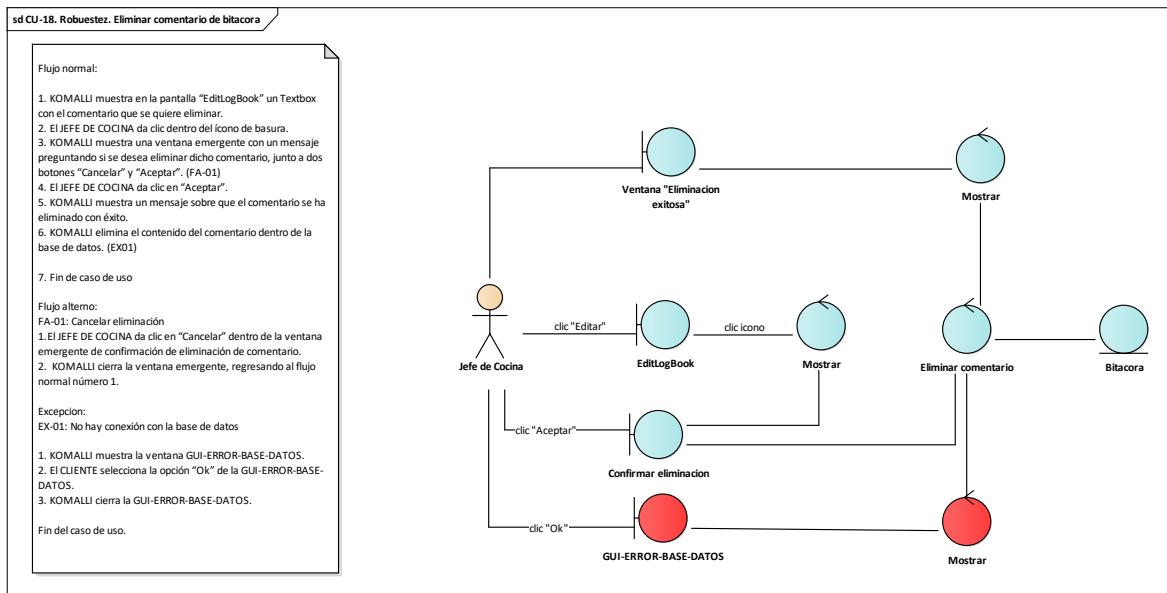


Ilustración 79 Diagrama de robustez CU-18

3.18.4 Diagrama de secuencia

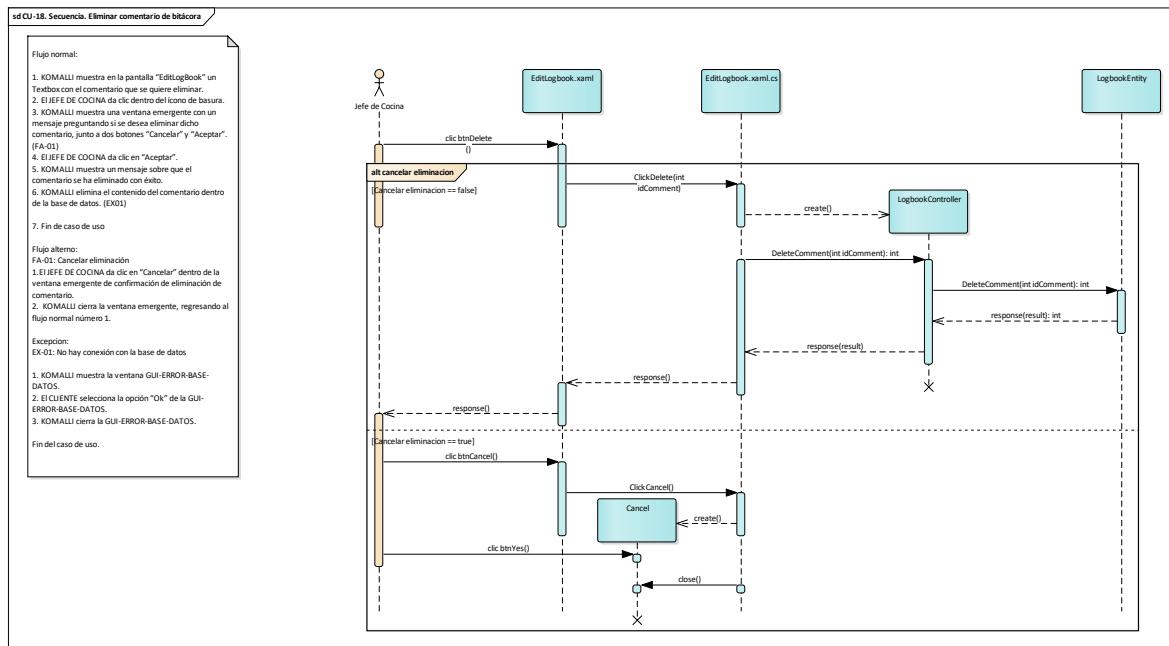


Ilustración 80 Diagrama de CU-18

3.19 CU 19 Generar Pdf de la bitacora

3.19.1 Prototipo

El prototipo muestra la interfaz de usuario de la aplicación Komalli. En la parte superior izquierda, hay un menú lateral con el logo Komalli y las siguientes opciones: Usuarios, Reportes (seleccionado), Inventario, Bitácora y Movimientos diarios. En la parte inferior del menú lateral, se muestra la foto y el nombre del usuario actual: Miriam Ramírez Zárate.

En la parte central superior, se titula "Reporte de bitácora". A la derecha, hay un botón "Descargar" y dos filtros: "Filtrar por: Fecha" y "Filtrar por: Incidentes".

Dentro de la sección principal, se titula "Reporte de bitácora" y muestra una tabla con los siguientes datos:

Nombre	Comentario	Fecha	Rol
Alejandro Sánchez Marín	Comentario...	01/03/2024	Personal de cocina
Miriam Ramírez Zárate	Comentario...	06/03/2024	Cajero
Ares Judda Rivera Soto	Comentario...	29/02/2024	Cajero
Paloma Osiris Báez Lara	Comentario...	05/03/2024	Jefe de cocina

En la parte inferior de la tabla, se indica la página "1".

Ilustración 81 Prototipo CU-19

3.19.2 Descripción

ID:	CU-19
Nombre del CU:	Generar PDF de la bitácora
Requisito funcional:	RF-08: Se deben generar informes de inventario en formato PDF.
Descripción:	Se crea un reporte con la información resumida de los comentarios realizados por los empleados.
Actor(es):	Gerente (Primario)
Disparador:	El Gerente da clic en el ícono de reporte para ingresar a “ LogbookReport ”.
Precondiciones:	PRE-01: Existe registrado en la bitácora al menos un comentario.
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI muestra la ventana “LogbookReport” con un filtro para crear el reporte. (ver EX-01) 2. El GERENTE selecciona un elemento del combobox para filtrar por <i>fecha</i> . 3. KOMALLI recupera de la base de datos los comentarios registrados de la bitácora y habilita el botón de “Descargar”. 4. KOMALLI muestra una previsualización del reporte. 5. El GERENTE selecciona la opción “Descargar” de “LogbookReport”. 6. KOMALLI guarda el reporte en el dispositivo y muestra la <u>GUI-REPORTE-GUARDADO</u>. 7. El GERENTE selecciona la opción “Continuar” de la <u>GUI-REPORTE-GUARDADO</u>. 8. KOMALLI cierra la <u>GUI-REPORTE-GUARDADO</u>. 9. Termina el caso de uso.
Flujos Alternos:	N/A
Excepciones:	EX-01: No hay conexión con la base de datos <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El GERENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01: Un nuevo reporte es generado y guardado en el dispositivo del gerente .
Incluye:	N/A
Extiende:	N/A

Tabla 19. Descripción CU-19

3.19.3 Diagrama de robustez

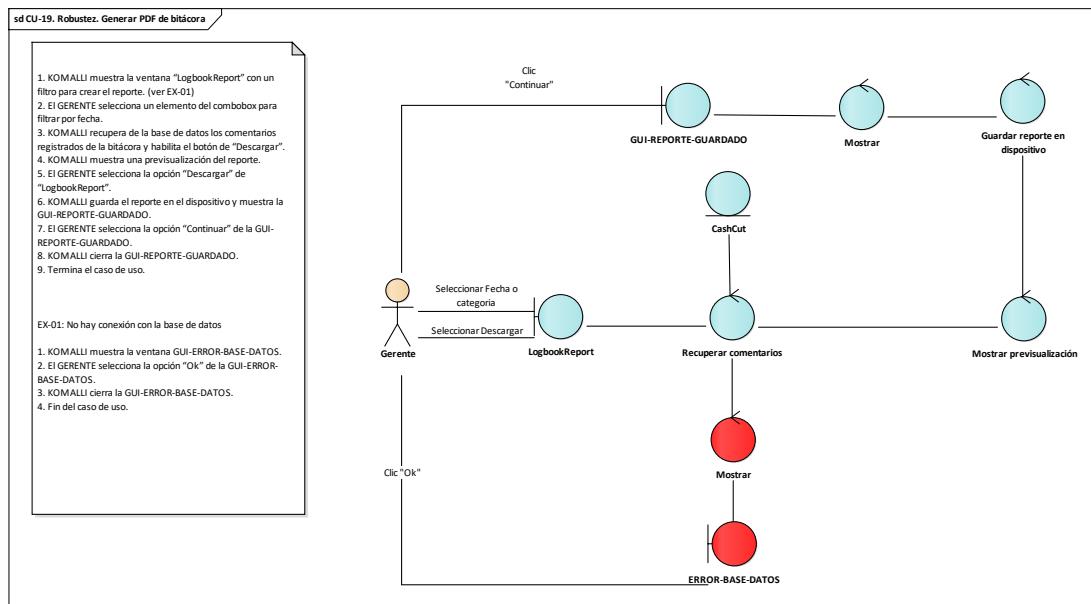


Ilustración 82 Diagrama de robustez CU-19

3.19.4 Diagrama de secuencia

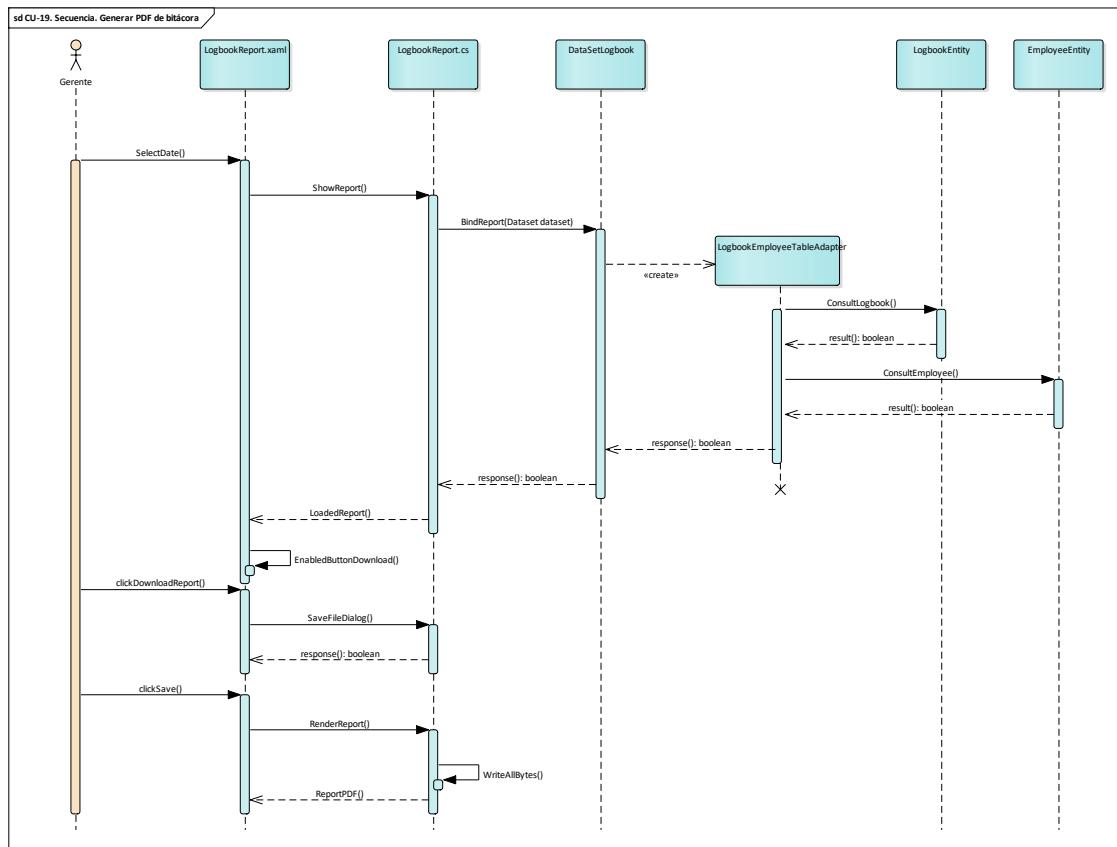


Ilustración 83 Diagrama de secuencia CU-19

3.20 CU 20 relizar cobro

3.20.1 Prototipo



The screenshot shows a mobile application interface for 'Realizar cobro'. The top navigation bar has a blue header with the 'Komalli' logo and a search bar. Below the header is a sidebar with a dark teal background containing the 'Caja' section (with 'Realizar pedido', 'Registrar gasto', and 'Realizar corte' options) and the 'Bitacora' section (with 'Agregar comentario' option). At the bottom of the sidebar is a user profile icon labeled 'Miriam Ramirez Zárate'. The main content area has a light green background. It features a title 'Realizar cobro' in a dark blue header. Below it is a table showing the items and their details:

Platillo	Cantidad	Precio unitario	Total
Desayuno	2	\$ 30	\$ 60
Jugo boing	1	\$ 19	\$ 19
Pay de limón	2	\$ 31	\$ 62

At the bottom of this section is a box containing 'TOTAL A PAGAR: \$ 141.00'.

To the right of the table is a green box with the text 'Total a pagar: \$ 141.00'. Below it is a field labeled 'Recibe:' followed by a text input box containing 'MXN'.

Ilustración 84 Prototipo CU-20



This screenshot shows the same mobile application interface as the previous one, but with additional fields for entering payment details. The sidebar and table remain the same. To the right of the table, there is a green box with the total amount '\$ 141.00'. Below it, there are three input fields: 'Recibe:' with the value '200' and 'MXN', 'Cambio:' with the value '59.00' and 'MXN', and 'Nombre:' with the value 'Paloma Osiris Báez'. At the bottom right is a blue button labeled 'Confirmar'.

Ilustración 85 Prototipo CU-20



FONDO DE EMPRESAS DE LA UNIVERSIDAD
VERACRUZANA

NOTA DE VENTA

Caj2864 - Alejandro Sánchez
04/05/2024 14:25:53

Cant.	Producto	Total
-------	----------	-------

1	Desayuno Estudiante	30.00
---	---------------------	-------

Total a pagar : 30.00
Efectivo : 100.00
Cambio : 70.00

1 Productos en 1 Movimientos

Comentario o sugerencia 228 814 2337
visite contacto@comedoresuv.com
REGIMEN GENERAL DE LEY PERSONAS MORALES

Ilustración 86 Prototipo Ticket

3.20.2 Descripción

ID:	CU- 20
Nombre del CU:	Realizar cobro
Requisito funcional:	RF-18: Se debe considerar el registro de ingresos diarios indicando la fecha y el monto recibido.
Descripción:	Se registra el cobro de un pedido realizado ya sea desde kiosko o desde caja.
Actor(es):	Cajero (Primario)
Disparador:	El Cajero da clic en la opción “Cobrar”.
Precondiciones:	PRE-01: Existe al menos un pedido registrado en la base de datos
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI recupera los detalles del pedido, mostrando el platillo, cantidad, precioUnitario, total, totalPagar en la ventana “MakeCollection”. (ver EX01). 2. El CAJERO ingresa el monto recibido por el cliente y da un “enter”. (ver FA-01). 3. KOMALLI calcula el cambio a regresar de acuerdo con el total a pagar y el monto recibido que registro el CAJERO, y muestra un campo para poner el nombre del que pertenece el pedido. 4. El CAJERO ingresa el nombre y da clic en la opción “Continuar” de la ventana “MakeCollection”. 5. KOMALLI modifica el estado del pedido a pagado en la base de datos y muestra la <u>GUI-PEDIDO-EXITOSO</u>. (ver EX01). 6. El CAJERO da clic en la opción “Ok” de la <u>GUI-PEDIDO-EXITOSO</u>. 7. KOMALLI cierra la <u>GUI-PEDIDO-EXITOSO</u>. 8. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Pedido desde kiosko</p> <ol style="list-style-type: none"> 1. KOMALLI calcula el cambio a regresar de acuerdo con el total a pagar y el monto recibido que registro el CAJERO y recupera el nombre del que pertenece el pedido y los muestra en la ventana “MakePayment”. 2. El CAJERO da clic en la opción “Continuar” de la ventana “MakePayment”. 3. Regresa al paso 5 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El CAJERO selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01. El estado del pedido es modificado a pagado.
Incluye:	N/A
Extiende:	N/A

Tabla 20 Descripción CU-20

3.20.3 Diagrama de robustez

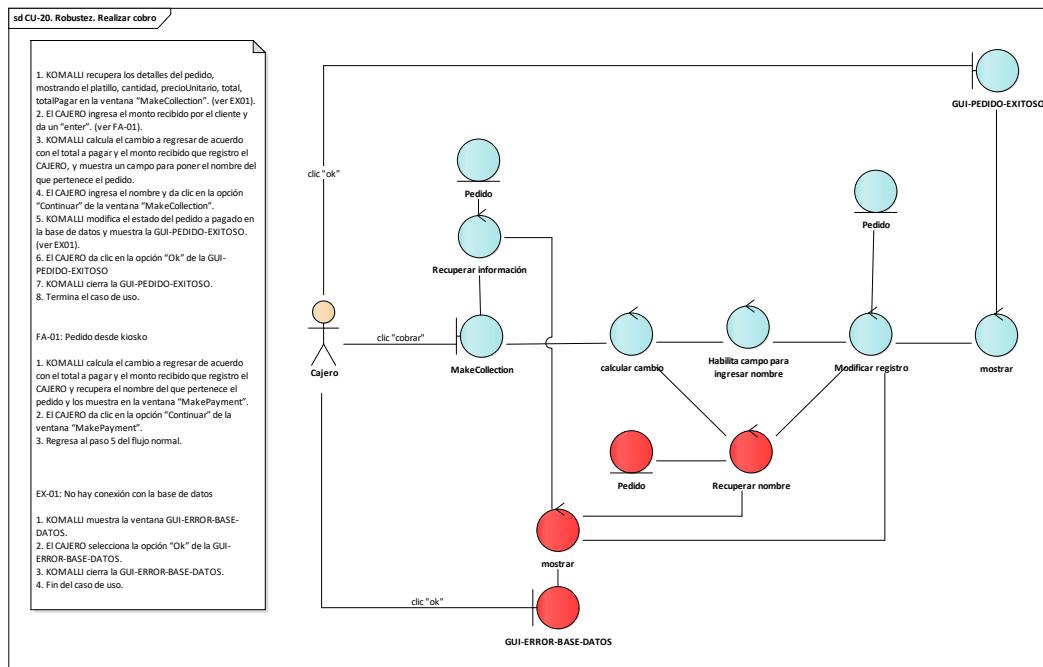


Ilustración 87 Diagrama Robustez CU-20

3.20.4 Diagrama de secuencia

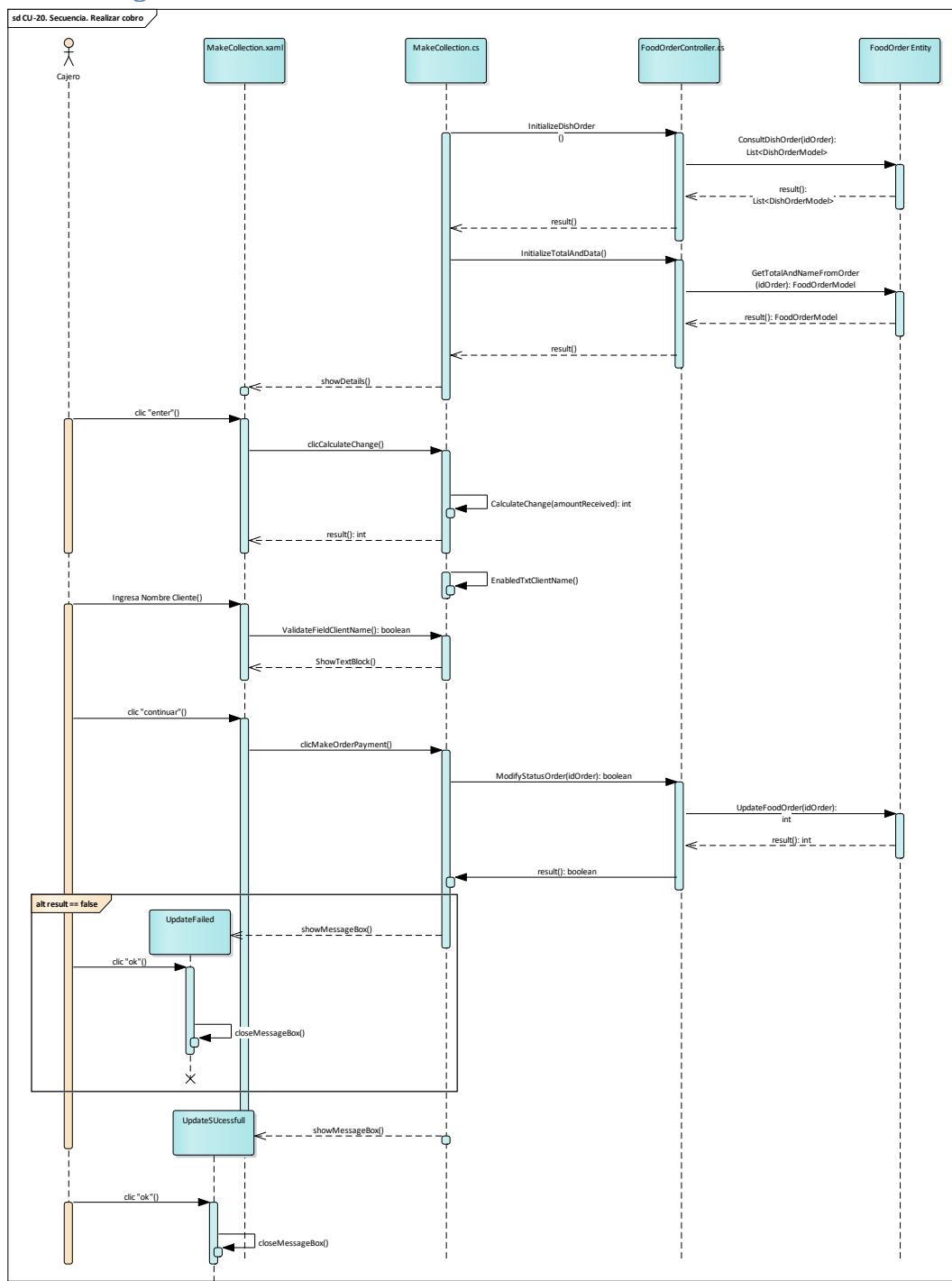


Ilustración 88 Diagrama secuencia CU-20

3.21 CU 21 Consultar pedidos kiosko

3.21.1 Prototipo

El prototipo muestra la interfaz de usuario de la aplicación Komalli para consultar pedidos de kiosco. La pantalla tiene un diseño moderno con un fondo verde claro. En la parte superior izquierda, hay un logo de Komalli (tres cuadrados azules y verdes) y el nombre "Komalli". A la derecha del logo, se encuentra el título "Pedidos de Kiosko" en un cuadro oscuro. Debajo del título, hay un campo de búsqueda con la placeholder "Buscar pedido" y un icono de lupa.

A la izquierda de la lista de pedidos, hay un menú lateral con las siguientes opciones:

- Caja ▾
 - Realizar pedido
 - Consultar pedidos** (destacada en azul)
 - Registrar gasto
 - Realizar corte
- Bitácora ▾
 - Agregar comentario

Dentro del cuadro principal, se muestra una tabla que detalla los pedidos registrados:

No. Pedido	Nombre	Cant. Platillo	Total	Opciones
Kio484	Miriam Ramírez Zárate	1	30	••• Cobrar
Kio982	Paloma Osiris Báez	2	48	•••
Kio105	Alejandro Sánchez Marín	4	147	•••
Kio487	Ares Judda Rivera Soto	2	100	•••
Kio007	Rosa Ysela Lara Landa	3	92	•••

Ilustración 89 Prototipo CU-21

3.21.2 Descripción

ID:	CU- 21
Nombre del CU:	Consultar pedidos kiosko
Requisito funcional:	RF-13: El sistema debe permitir consultar los pedidos realizados en el kiosko y proceder a su cobro.
Descripción:	Se consulta la lista de pedidos realizados desde el kiosko
Actor(es):	Cajero (Primario)
Disparador:	El Cajero da clic en “Consultar pedidos” de la sección Caja del menú del sistema.
Precondiciones:	PRE-01: Existe al menos un pedido registrado en el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI recupera de la base de datos los detalles de los pedidos realizados en kiosko registrados en el sistema, mostrando el numeroPedido, nombrePedido en la ventana “ConsultOrder”. (ver EX01). 2. El CAJERO ingresa un número de pedido dentro del buscador de la ventana “ConsultOrder” y da clic en ícono de buscar (una lupa). 3. KOMALLI busca aquellos números de pedidos que coincidan con el número ingresado en el buscador y los muestra en forma de tabla en la ventana “ConsultOrder ”. 4. Termina el caso de uso.
Flujos Alternos:	N/A
Excepciones:	EX-01: No hay conexión con la base de datos <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El CAJERO selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	N/A
Incluye:	N/A
Extiende:	CU-21. Realizar cobro

Tabla 21 Descripción CU-21

3.21.3 Diagrama de robustez

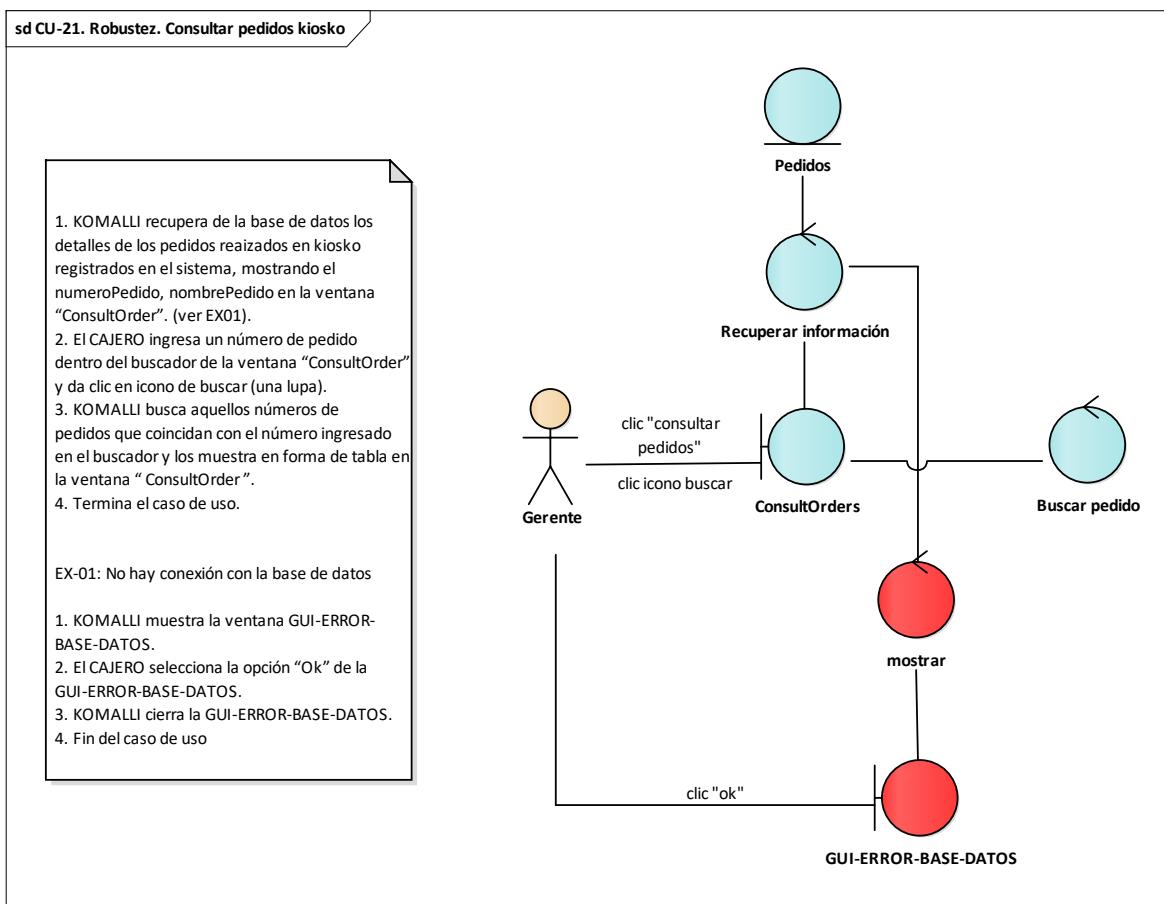


Ilustración 90 Diagrama de robustez CU-21

3.21.4 Diagrama de secuencia

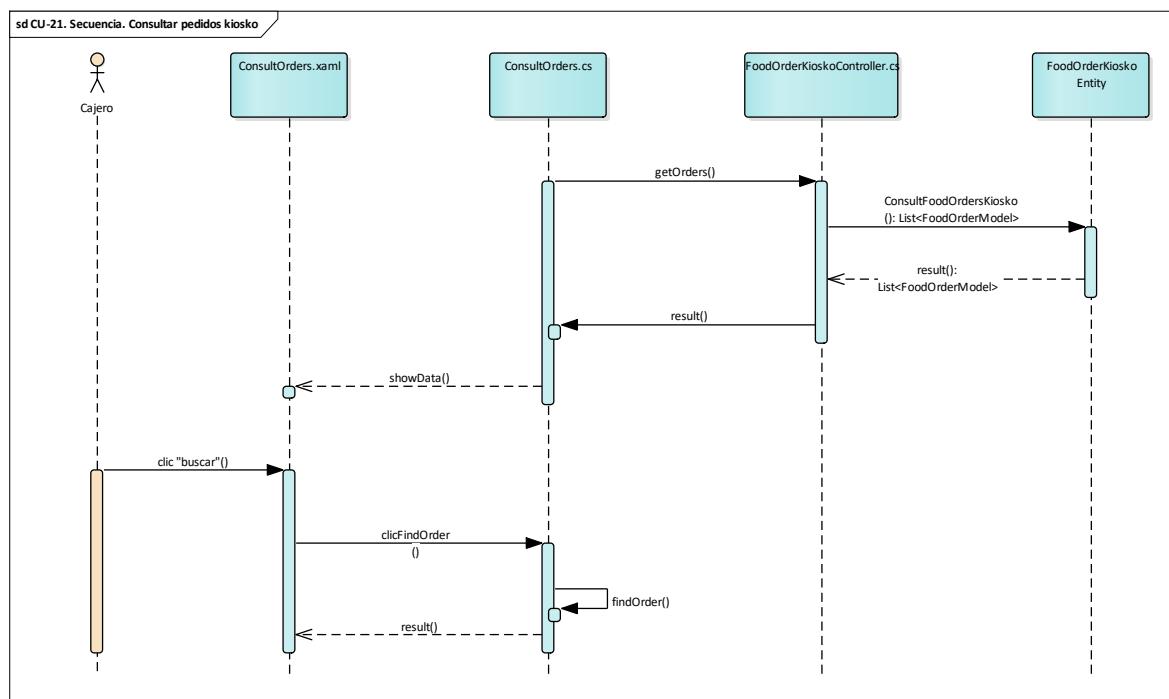


Ilustración 91 Diagrama secuencia CU-21

3.22 CU 22 Realizar corte de caja

3.22.1 Prototipo

El prototipo muestra la interface de Komalli para el corte de caja. En la parte izquierda, un sidebar incluye el logo de Komalli, una barra superior con "Caja", "Realizar cobro", "Registrar gasto", "Realizar corte", "Bitacora" y "Agregar comentario", y un icono de perfil de Miriam Ramírez Zárate.

Corte de caja

Saldo inicial	Total de entradas	Total de salidas	Balance
\$510.00	\$210.00	\$124.00	\$596.00

Fecha de corte: 19/03/2024 Saldo inicial para el siguiente día: **Realizar corte**

Historial de ventas

N. Operación	Fecha y hora	Nombre del cliente	N. de platillos	Subtotal
1120	19/03/2024-3:25:03	Alejandro Sánchez Marín	2	\$60
1121	19/03/2024-3:27:22	Paloma Osiris Báez Lara	3	\$90
1122	19/03/2024-3:27:59	Ares Judda Rivera Soto	1	\$30
1123	19/03/2024-3:29:06	Miriam Ramírez Zárate	1	\$30

Ilustración 92 Prototipo CU-22

3.22.2 Descripción

ID:	CU-22
Nombre del CU:	Realizar corte de caja
Requisito funcional:	RF-19: Komalli debe permitir la realización de balances diarios para analizar los ingresos y gastos del día.
Descripción:	Se obtienen las ventas que se realizaron durante un día para realizar el corte de caja.
Actor(es):	Cajero (Primario)
Disparador:	El Cajero da clic en el ícono de caja para ingresar a “CashCut”.
Precondiciones:	<p>PRE-01: Se han realizado ventas durante un día.</p> <p>PRE-02: Existe por lo menos un saldo inicial registrado en la base de datos.</p>
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI obtiene los registros de las órdenes y actualiza el último registro de la base de datos a la fecha actual. (ver EX-01) 2. KOMALLI obtiene el último registro del balance inicial de la base de datos y lo muestra en la ventana, además calcula el total de entradas y salidas de la fecha actual. (ver EX-01) 3. El CAJERO ingresa el <i>saldoInicial</i> que habrá para el siguiente día de ventas. 4. KOMALLI valida la entrada y habilita el botón de “Realizar corte”. (ver FA-01) 5. El CAJERO selecciona la opción “Realizar corte”. 6. KOMALLI guarda en la base de datos el <i>saldoInicial</i> así como el <i>totalEntradas</i>, <i>totalSalidas</i> y el <i>balance</i>. (ver EX-01) 7. KOMALLI muestra la <u>GUI-CORTECAJA-REGISTRADO</u>. 8. El CAJERO selecciona la opción “Continuar” de la <u>GUI-CORTECAJA-REGISTRADO</u>. 9. KOMALLI cierra la <u>GUI-CORTECAJA-REGISTRADO</u>. 10. KOMALLI termina el caso de uso. 11. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: Campos inválidos</p> <ol style="list-style-type: none"> 1. KOMALLI detecta campos inválidos y muestra la alerta <u>GUI-DATOS-INVALIDOS</u>. 2. El CAJERO da clic en la opción “Ok” de la alerta <u>GUI-DATOS-INVALIDOS</u>. 3. KOMALLI cierra alerta <u>GUI-DATOS-INVALIDOS</u>. 4. Regresar al paso 3 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El CAJERO selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01: Un nuevo corte de caja es registrado en la base de datos.
Incluye:	N/A
Extiende:	N/A

Tabla 22. Descripción CU-22

3.22.3 Diagrama de robustez

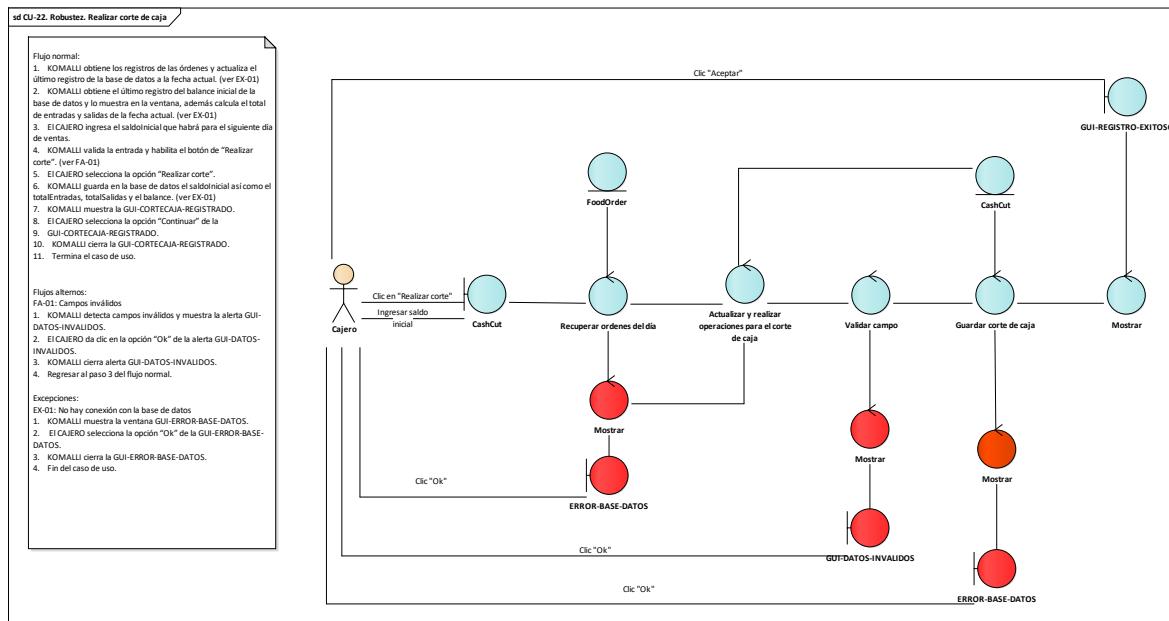


Ilustración 93 Diagrama robustez CU-22

3.22.4 Diagrama de secuencia

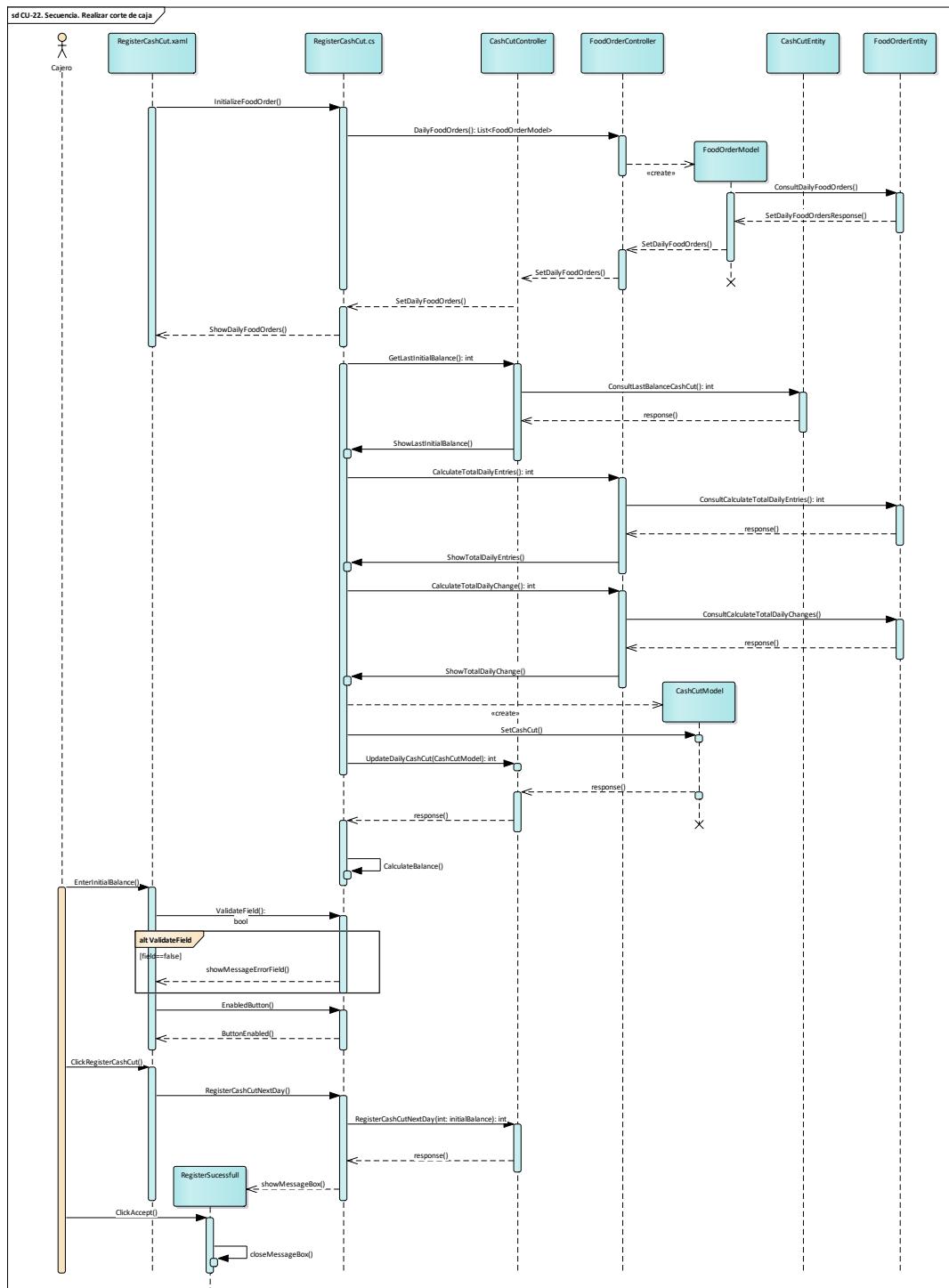


Ilustración 94 Diagrama secuencia CU-22

3.23 CU 23 Generar Pdf de movimientos diarios

3.23.1 Prototipo

El prototipo muestra una interfaz web para 'Movimientos diarios'. En la parte superior izquierda, hay un logo 'Komalli' y un menú lateral con opciones: 'Usuarios', 'Reportes' (seleccionado) y 'Movimientos diarios'. La parte central tiene un encabezado 'Movimientos diarios' y botones para 'Descargar' y 'Historial de ventas'. Los botones para 'Fecha inicio' y 'Fecha fin' tienen un icono de flecha que indica que se puede filtrar por fecha. Abajo, se muestra un cuadro titulado 'Historial de ventas' con un balance total de '\$596.00'. Se detallan las operaciones realizadas:

N. Operación	Fecha y hora	Nombre del cliente	N. platillos	Subtotal
1120	19/03/2024-3:25:03	Alejandro Sánchez Marín	2	\$60
1121	19/03/2024-3:27:22	Paloma Osiris Báez Lara	3	\$90
1122	19/03/2024-3:27:59	Ares Judda Rivera Soto	1	\$30
1123	19/03/2024-3:29:06	Miriam Ramírez Zárate	1	\$30

Ilustración 95 Prototipo CU-23

3.23.2 Descripción

ID:	CU-23
Nombre del CU:	Generar PDF de movimientos diarios
Requisito funcional:	RF-08: Se deben generar informes de inventario en formato PDF.
Descripción:	Se crea un reporte con la información resumida de los movimientos diarios realizados en un día.
Actor(es):	Gerente (Primario)
Disparador:	El Gerente da clic en el ícono de reporte para ingresar a “ DailyTransactionsReport ”.
Precondiciones:	PRE-01: Existe al menos un corte de caja registrado en el sistema.
Flujo Normal	<ol style="list-style-type: none">KOMALLI muestra la ventana “DailyTransactionsReport” con un filtro para crear el reporte.El GERENTE selecciona un elemento del combobox para filtrar por <i>fecha</i>.KOMALLI recupera de la base de datos los cortes de caja registrados, así como los pedidos del día y habilita el botón de “Descargar”. (ver EX-01)KOMALLI muestra una previsualización del reporte.El GERENTE selecciona la opción “Descargar” de “DailyTransactionsReport”.KOMALLI guarda el reporte en el dispositivo y muestra la <u>GUI-REPORTEGUARDADO</u>.El GERENTE selecciona la opción “Continuar” de la

	<u>GUI-REPORTE-GUARDADO.</u> 8. KOMALLI cierra la <u>GUI-REPORTE-GUARDADO</u> . 9. Termina el caso de uso.
Flujos Alternos:	N/A
Excepciones:	EX-01: No hay conexión con la base de datos 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u> . 2. El GERENTE selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u> . 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u> . 4. Fin del caso de uso.
Postcondiciones:	POST-01: Un nuevo reporte es generado y guardado en el dispositivo del gerente .
Incluye:	N/A
Extiende:	N/A

Tabla 23. Descripción CU-23

3.23.3 Diagrama de robustez

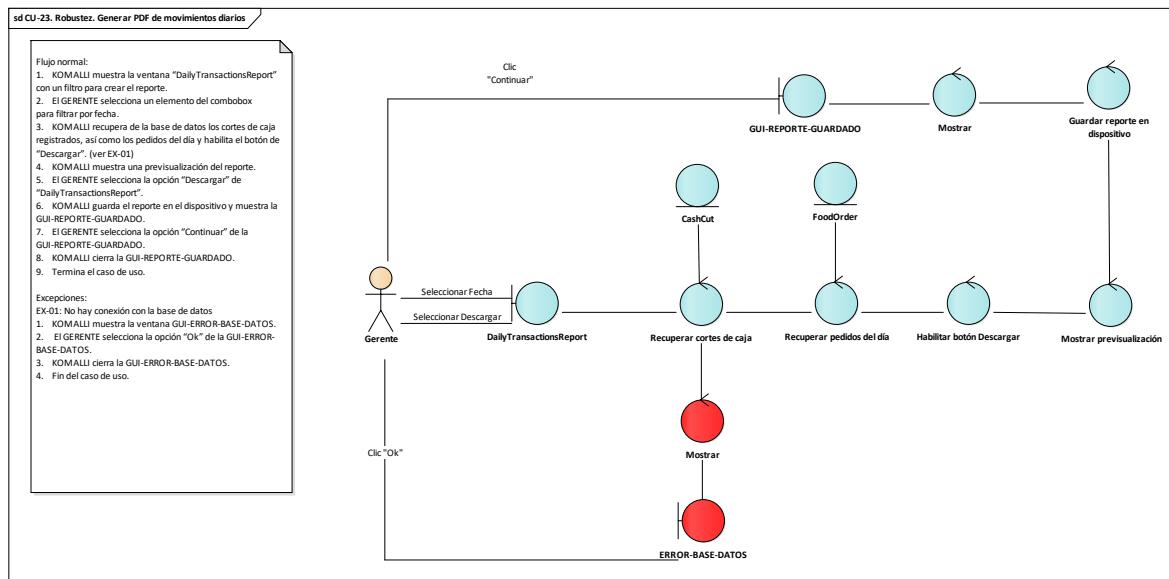


Ilustración 96 Diagrama robustez CU-23

3.23.4 Diagrama de secuencia

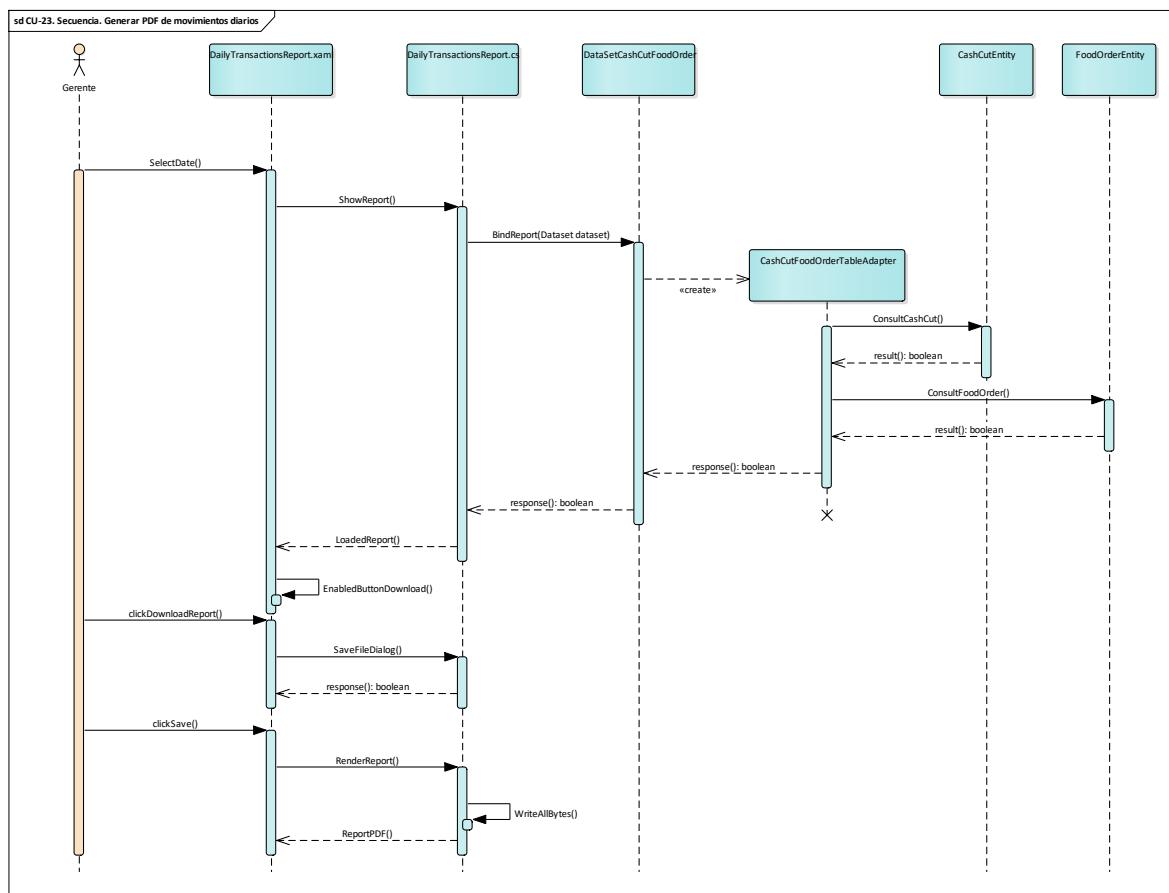


Ilustración 97 Diagrama secuencia CU-23

3.24 CU 24 cancelar pedido

3.24.1 Prototipo

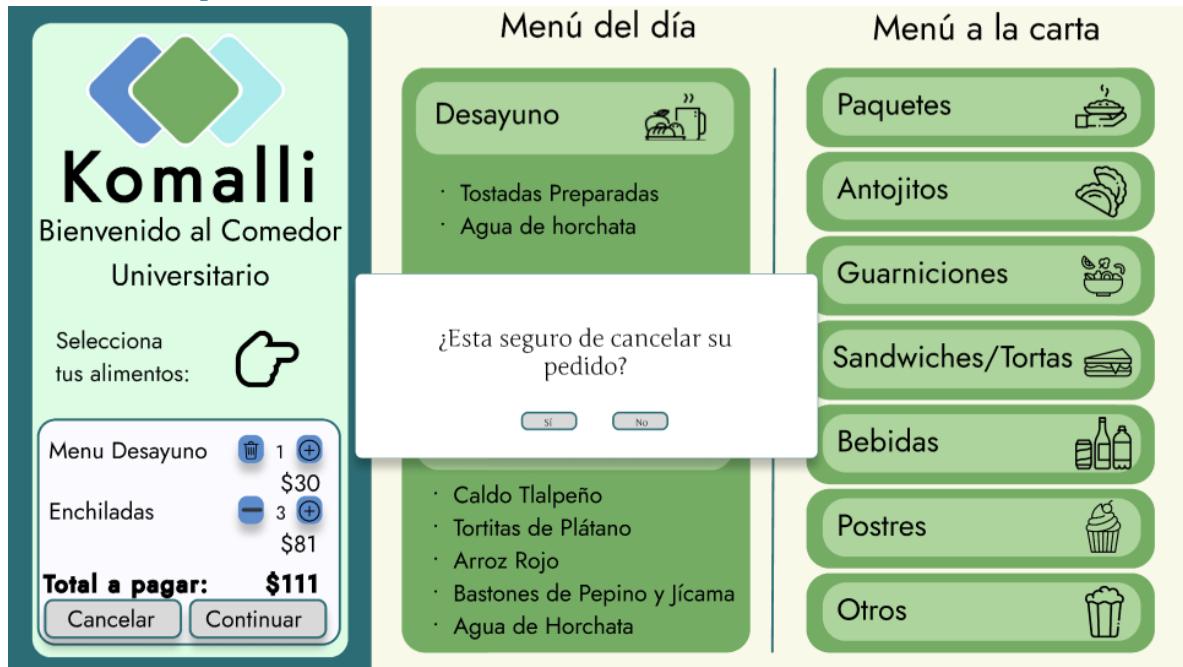


Ilustración 98 Prototipo CU-24

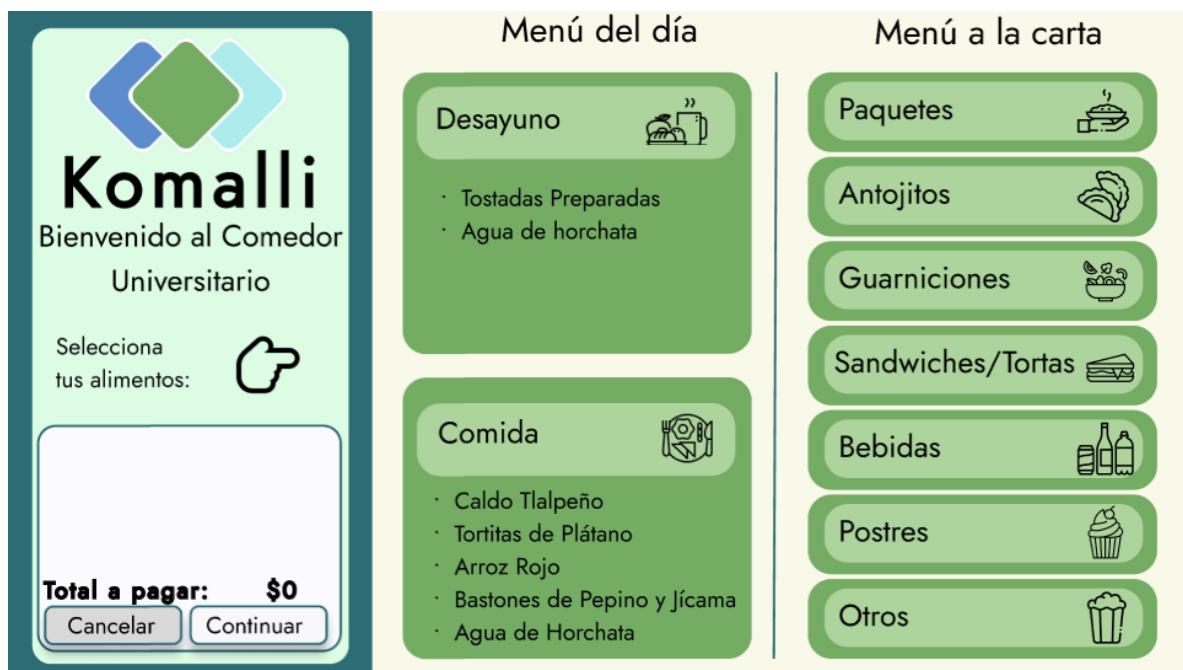


Ilustración 99 Prototipo CU-24

3.24.2 Descripción

ID:	CU-24
Nombre del CU:	Cancelar pedido
Requisito Funcional:	RF-11: El sistema debe permitir a los usuarios editar pedidos existentes, agregar o eliminar productos y modificar cantidades.
Descripción:	Se desea quitar los pedidos realizados en el kiosko.
Actor(es):	Cliente (Primario)
Disparador:	El Cliente desea cancelar su pedido .
Precondiciones:	PRE-01: Existe por lo menos un pedido en el listado de pedidos.
Flujo Normal:	<ol style="list-style-type: none"> 1. KOMALLI muestra un listado de los pedidos que se han realizado. 2. El CLIENTE selecciona la opción “Cancelar” de la ventana “HomeKiosko”. 3. KOMALLI despliega la <u>GUI-CANCELAR-PEDIDO</u>. 4. El CLIENTE selecciona la opción “Sí”. 5. KOMALLI cierra la <u>GUI-CANCELAR-PEDIDO</u>. 6. KOMALLI borra todos los pedidos que existían en la lista de pedidos y cambia el subtotal a \$0. 7. Fin del caso de uso.
Flujos Alternos:	<p>FA-01: Selecciona “No” al cancelar pedido.</p> <ol style="list-style-type: none"> 1. El CLIENTE selecciona la opción “No” de la <u>GUI-CANCELAR-PEDIDO</u>. 2. KOMALLI cierra la <u>GUI-CANCELAR-PEDIDO</u>. 3. Fin del caso de uso.
Excepciones:	N/A
Postcondiciones:	<p>POST-01: Se borran los pedidos previamente realizados.</p> <p>POST-02: El subtotal de los pedidos se cambia a \$0.</p>
Incluye:	N/A
Extiende:	N/A

Tabla 24. Descripción CU-24

3.24.3 Diagrama de robustez

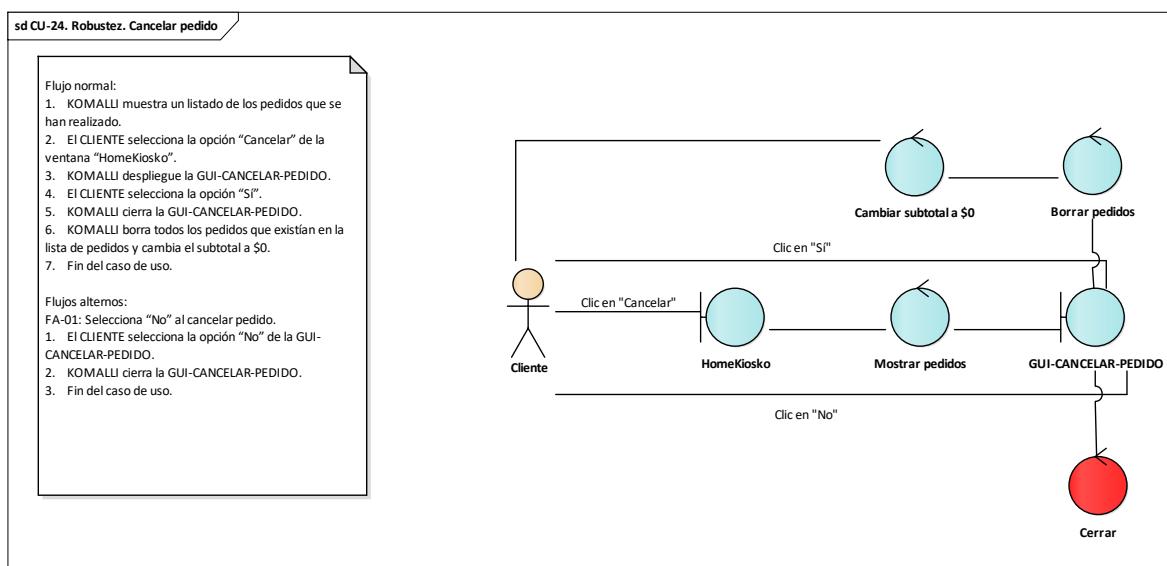


Ilustración 100 Diagrama de robustez CU-24

3.24.4 Diagrama de secuencia

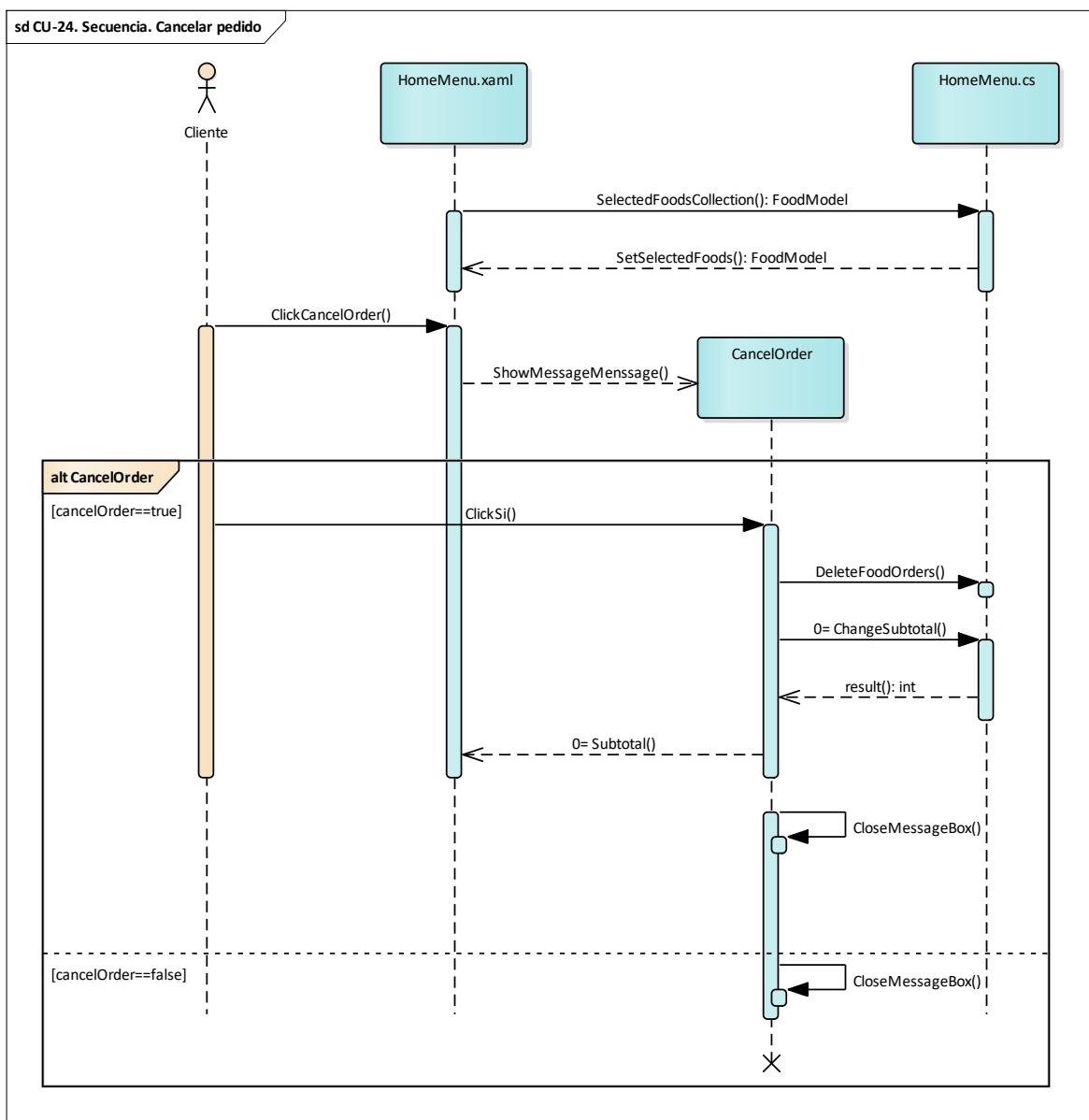


Ilustración 101 Diagrama de secuencia CU-24

3.25 CU 25 consultar menús registrados

3.25.1 Prototipo



Ilustración 102 Prototipo CU-25



Ilustración 103 Prototipo CU-25



Ilustración 104 Prototipo CU-25

3.25.2 Descripción

ID:	CU-25
Nombre del CU:	Consultar menús registrados
Requisito funcional:	RF-16: Se debe permitir a los empleados planificar y organizar los menús semanales.
Descripción:	Se busca que el personal de cocina pueda consultar la información sobre los menús registrados, según su día de interés
Actor(es):	Personal de cocina (primario)
Disparador:	El PERSONAL DE COCINA selecciona la opción de ver menú dentro de la sección de menú, del home correspondiente
Precondiciones:	PRE-01.- El usuario debe estar registrado en el sistema PRE-02.- El usuario debe de tener su perfil habilitado dentro del sistema. PRE-03.- El usuario debe contar con los permisos necesarios para realizar la operación (ser Personal de cocina).
Flujo Normal	<ol style="list-style-type: none"> KOMALLI muestra un formulario la información del menú del día en curso con los campos: <i>Entrada, plato fuerte guarnición, ensalada, bebida, tipo de comida, fecha</i>, siendo (<i>plato fuerte, bebida, tipo y fecha</i>) campos obligatorios para el proceso de publicación. Y un calendario para filtrar las búsquedas según el día (Ver EX-01). El PERSONAL DE COCINA selecciona la fecha de la cual le interesa saber el menú. KOMALLI busca en la base de datos los menús que hagan match con la fecha ingresada. (Ver EX-01). KOMALLI muestra los menús obtenidos a través de la consulta. (Ver FA.01) Termina el caso de uso.
Flujos Alternos:	FA-01: Menús inexistentes.

	<ol style="list-style-type: none"> 1. KOMALLI muestra la tabla de resultados vacía, lo que se interpreta como que el sistema no ha encontrado ningún elemento que coincida con la búsqueda solicitada. 2. Regresa al paso 2 del flujo normal.
Excepciones:	EX-01: No hay conexión con la base de datos <ol style="list-style-type: none"> 1. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>. 2. El PERSONAL DE COCINA selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>. 3. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>. 4. Fin del caso de uso.
Postcondiciones:	POST-01: EL PERSONAL DE COCINA obtiene la información del menú de interés.
Incluye:	N/A
Extiende:	N/A

Tabla 25. Descripción CU-25

3.25.3 Diagrama de robustez

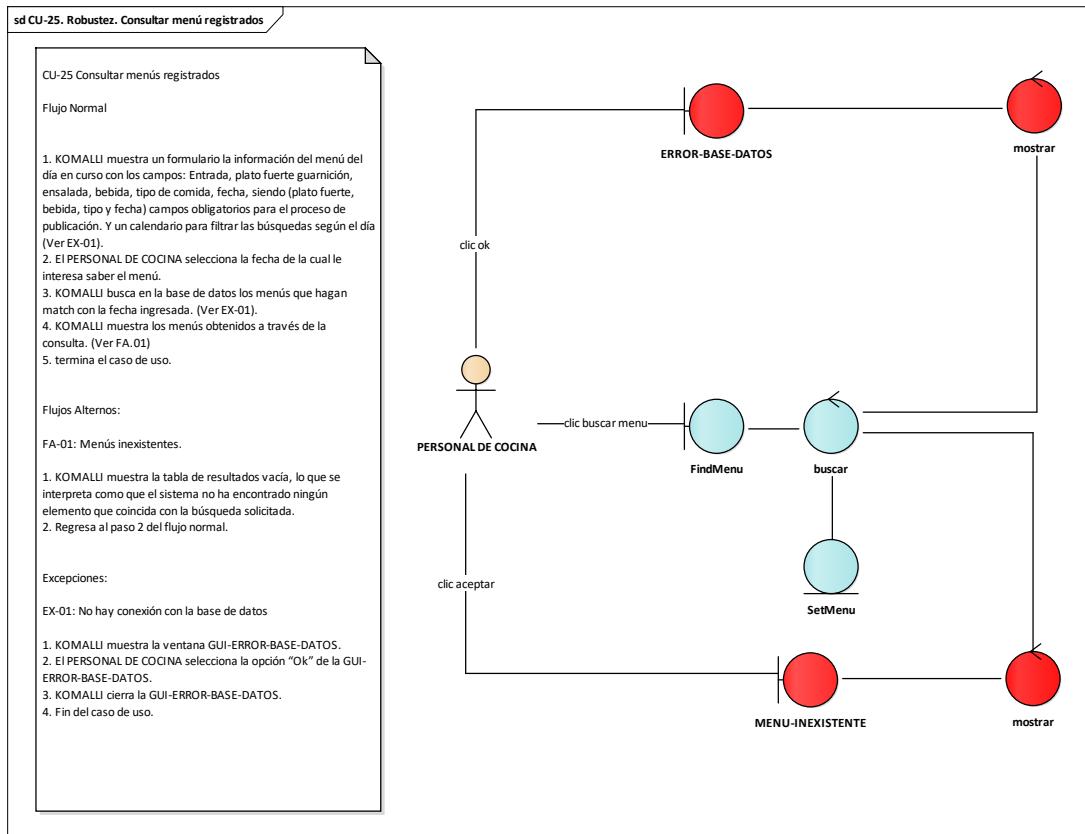


Ilustración 105 Diagrama de robustez CU-25

3.25.4 Diagrama de secuencia

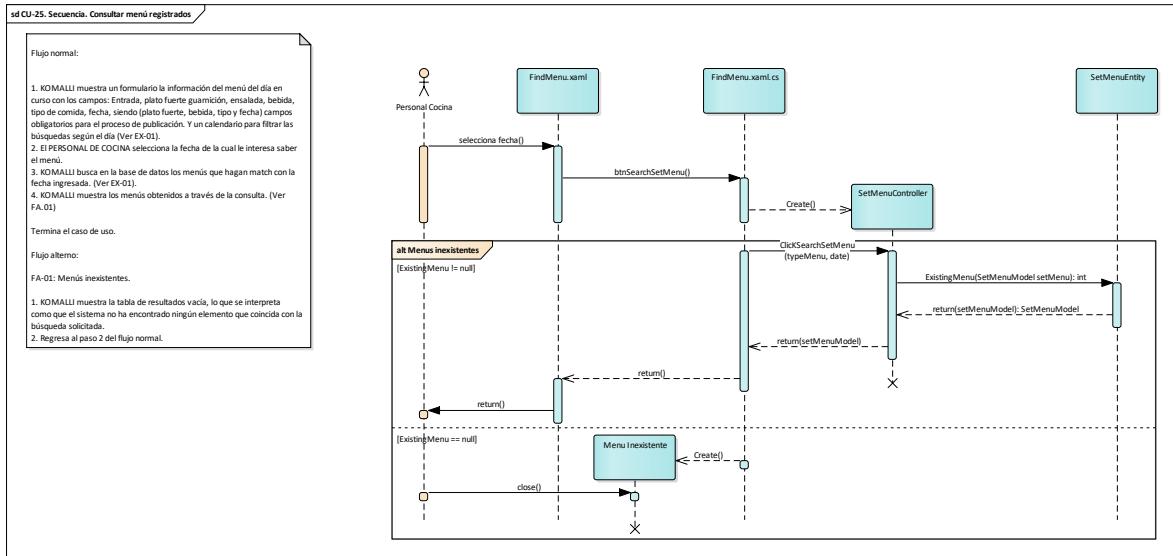


Ilustración 106 Diagrama de secuencia CU-25

3.26 CU 26 Consultar comentarios en bitácora

3.26.1 Prototipo

The screenshot shows the Komalli application interface. On the left is a sidebar with a logo, navigation links for 'Inventario', 'Bitácora' (which is currently selected), and 'Comentarios'. A user profile for 'Miriam Ramírez Zárate' is also visible. The main area is titled 'Bitácora' and shows a list of comments. Each comment includes the author's name, the type of report ('Merma', 'Incidencia', 'General', 'Otro'), and the date. The 'Bitácora' tab is highlighted in blue.

Author	Type	Date
Ares Judda Rivera Soto	Merma	05 / 03 / 2024
Alejandro Sánchez Marín	General	05 / 03 / 2024
Paloma Osiris Báez Lara	Otro	05 / 03 / 2024
Alejandro Sánchez Marín	Otro	01 / 03 / 2024
Ares Judda Rivera Soto	Merma	29 / 02 / 2024

Ilustración 107 Prototipo CU-26



Ilustración 108 Prototipo CU-26

3.26.2 Descripción

ID:	CU-26
Nombre del CU:	Consultar comentarios en bitácora
Requisito funcional:	RF-16: Komalli permite administrar por medio de una bitácora aquellos sucesos o imprevistos que puedan suscitarse a lo largo de la jornada laboral dentro del comedor.
Descripción:	Permite al jefe de cocina poder consultar todos los comentarios realizados por él y sus compañeros de cocina.
Actor(es):	Jefe de Cocina (primario)
Disparador:	El JEFE DE COCINA selecciona una fecha en “Comentarios” de la sección Bitácora.
Precondiciones:	PRE-01.- El usuario debe contar con los permisos necesarios para realizar la operación.
Flujo Normal	<ol style="list-style-type: none"> 1. KOMALLI recupera los comentarios de la base de datos. 2. KOMALLI muestra la información de bitácora con la siguiente información: <i>Autor del comentario, fecha en la que se realizó, descripción.</i> campos obligatorios para el proceso de publicación. Y un calendario para filtrar las búsquedas según el día (Ver EX-01)(FA-01) 3. El JEFE DE COCINA visualiza los comentarios añadidos en la fecha seleccionada. 4. Termina el caso de uso.
Flujos Alternos:	<p>FA-01: No hay comentarios registrados</p> <ol style="list-style-type: none"> 1. KOMALLI muestra la tabla de resultados vacía. 2. Regresa al paso 2 del flujo normal.
Excepciones:	<p>EX-01: No hay conexión con la base de datos</p> <ol style="list-style-type: none"> 3. KOMALLI muestra la ventana <u>GUI-ERROR-BASE-DATOS</u>.

	<p>4. El PERSONAL DE COCINA selecciona la opción “Ok” de la <u>GUI-ERROR-BASE-DATOS</u>.</p> <p>5. KOMALLI cierra la <u>GUI-ERROR-BASE-DATOS</u>.</p> <p>6. Fin del caso de uso.</p>
Postcondiciones:	POST-01: El JEFE DE COCINA obtiene la información de los comentarios añadidos dentro de bitácora.
Incluye:	N/A
Extiende:	N/A

Tabla 26. Descripción CU-26

3.26.3 Diagrama de robustez

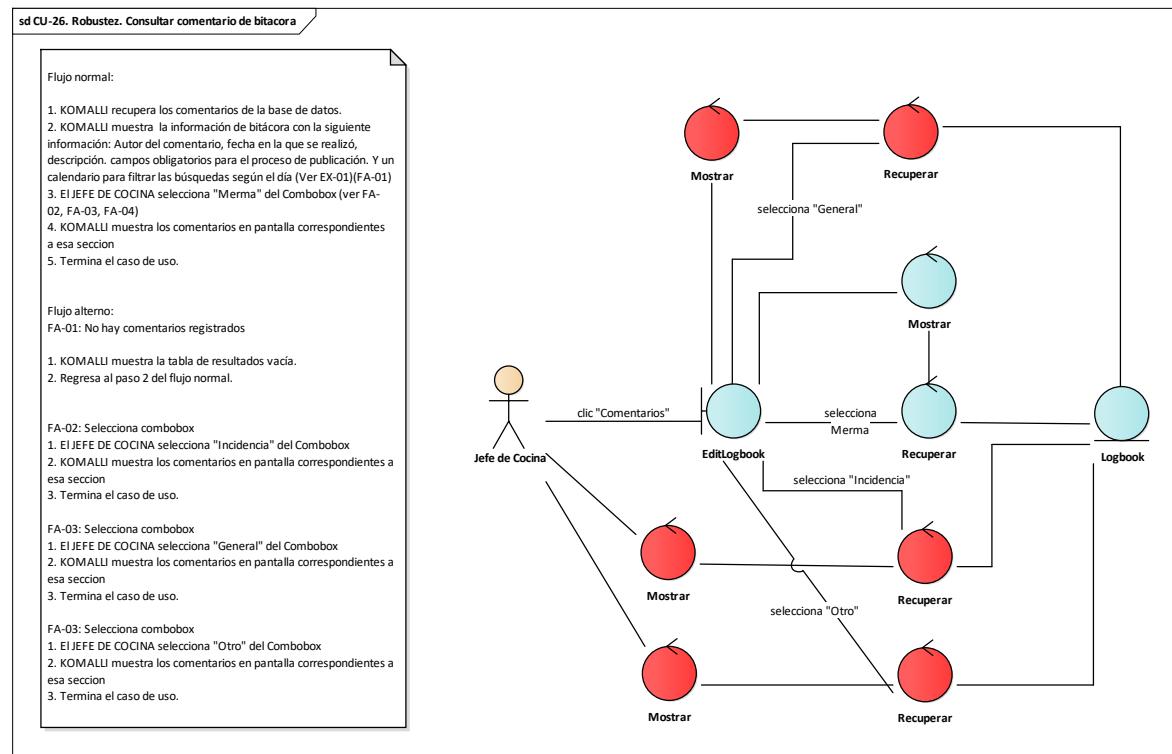


Ilustración 109 Diagrama robustez CU-26

3.26.4 Diagrama de secuencia

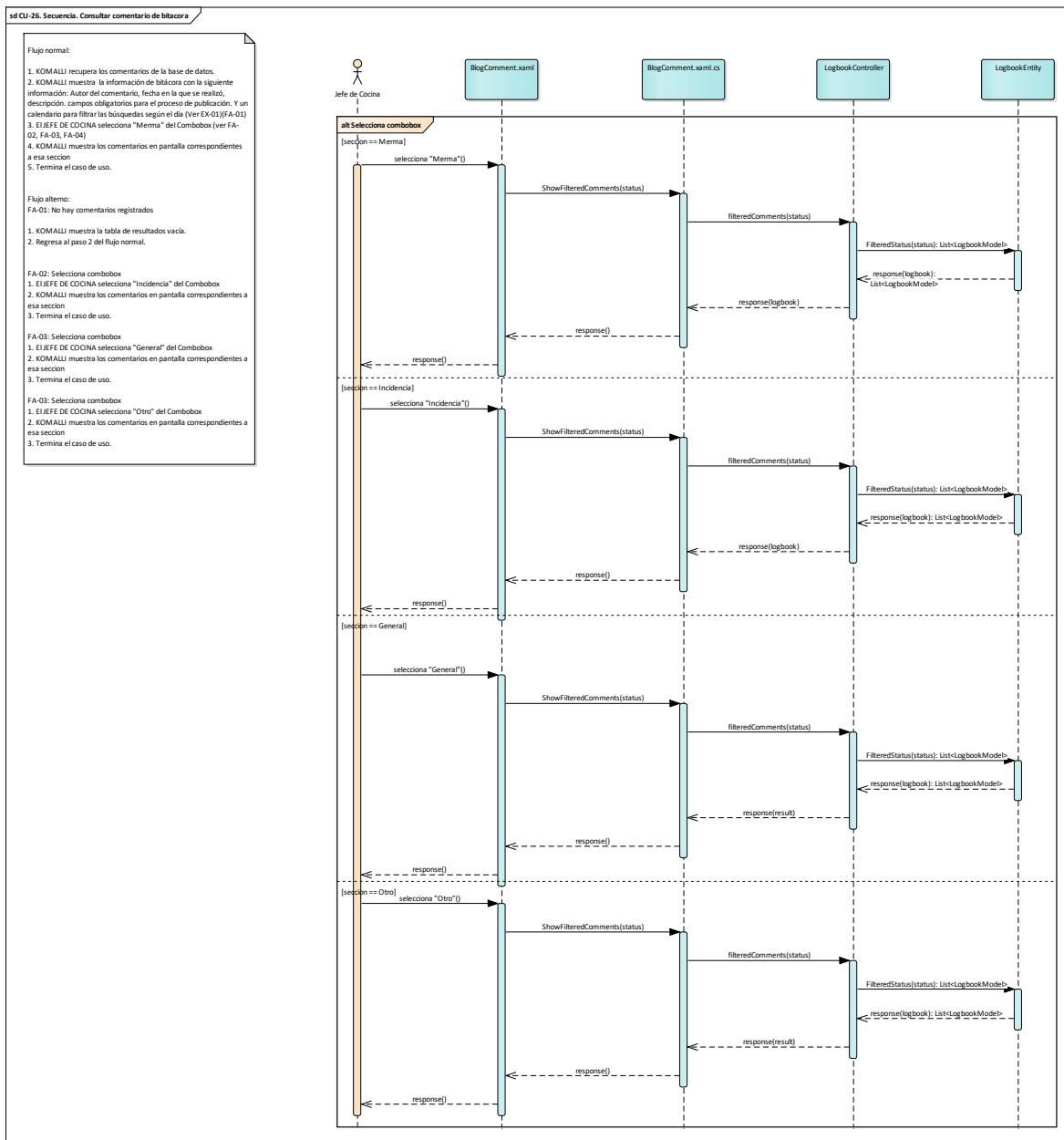


Ilustración 110 Diagrama secuencia CU-26

3.27 Modelo de Dominio

El diagrama de dominio describe las entidades y relaciones dentro del sistema Komalli. El comedor, que tiene atributos como capacidad, facultad, horarios de apertura y cierre, y nombre, emplea una lista de trabajadores, cada uno identificado por su celular, correo, nombre, número personal y rol (supervisor, cajero o cocinero). Los clientes, que pueden ser estudiantes (con matrícula) o personal de la UV (con departamento y número personal), realizan pedidos que incluyen una lista de comidas, detalladas por consumidor, encargado del pedido, fecha de consumo y total. Los ingredientes, especificados por cantidad, clave, código de barras, medida y nombre, son gestionados en un inventario que registra la fecha de reabasto y la lista de ingredientes. Las comidas cuentan con costo y nombre las cuales están asociadas a los pedidos. El personal de la UV se clasifica en áreas como académicos, administrativos y docentes.

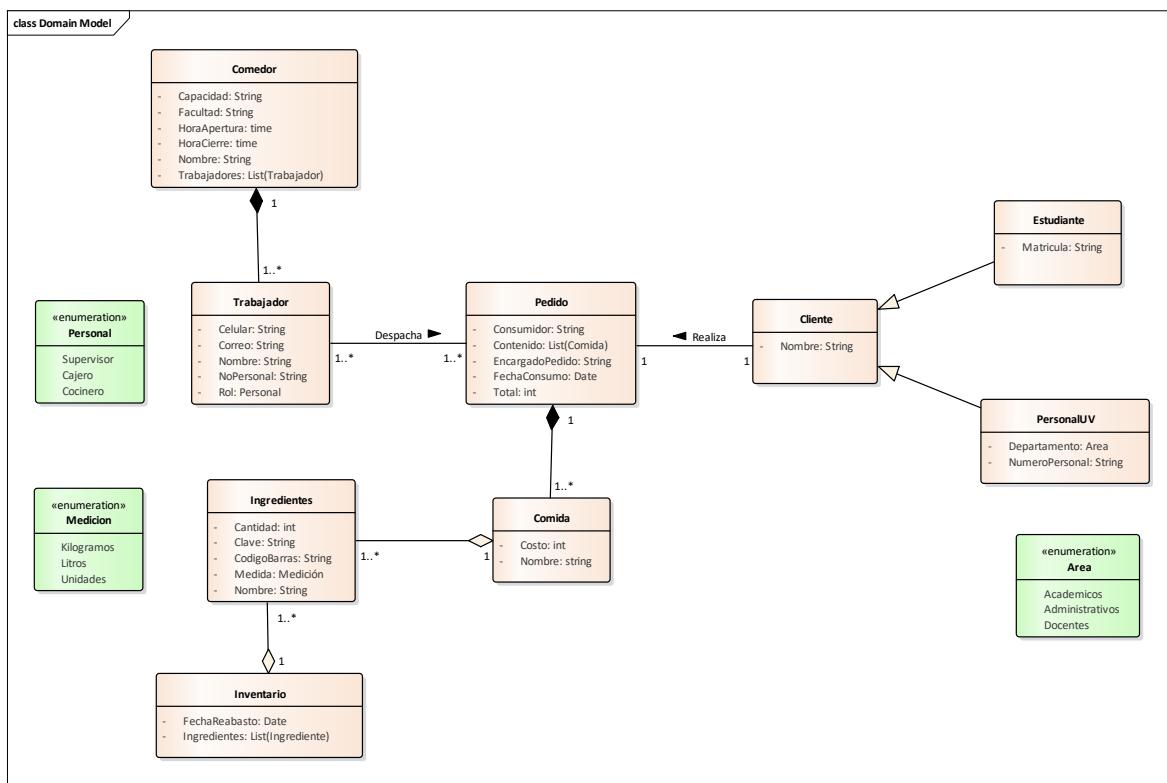


Ilustración 111 Diagrama del dominio

3.28 Arquitectura Técnica

La arquitectura técnica del sistema Komalli se organiza en tres capas principales: presentación, datos y lógica de aplicación, que facilitan la separación de responsabilidades y la gestión eficiente del sistema. La capa de presentación abarca las interfaces de usuario y controles reutilizables. La capa de datos maneja las clases de modelo, reglas de validación y utilidades necesarias para la persistencia y manipulación de la información. La capa de lógica de aplicación coordina los controladores que implementan la lógica de negocio, asegurando el correcto flujo de información y procesos. Adicionalmente, el despliegue del sistema se estructura entre una computadora cliente y un servidor, comunicándose a través del protocolo ODBC, y utilizando SQL Server para la gestión de la base de datos. Esta arquitectura modular permite una gestión eficiente de pedidos de comida, empleados y registros financieros, logrando así que el proyecto cuente con una gran robustez y escalabilidad en el funcionamiento del sistema.

3.28.1 Modelo de paquetes

Este diagrama de paquetes muestra la arquitectura de la aplicación Komalli en tres capas principales. La capa de presentación (View) incluye páginas XAML como BlogComment.xaml y RegisterUser.xaml, controles de usuario reutilizables como CommentCard.xaml, y ventanas principales como Login.xaml. La capa de datos (Model) contiene clases de modelo como EmployeeModel.cs y FoodOrderModel.cs, reglas de validación como EmailValidationRule.cs, y utilidades como LoggerManager.cs. La capa de lógica de aplicación (Controller) gestiona la lógica de negocio con controladores como CashCutController.cs y FoodOrderController.cs.

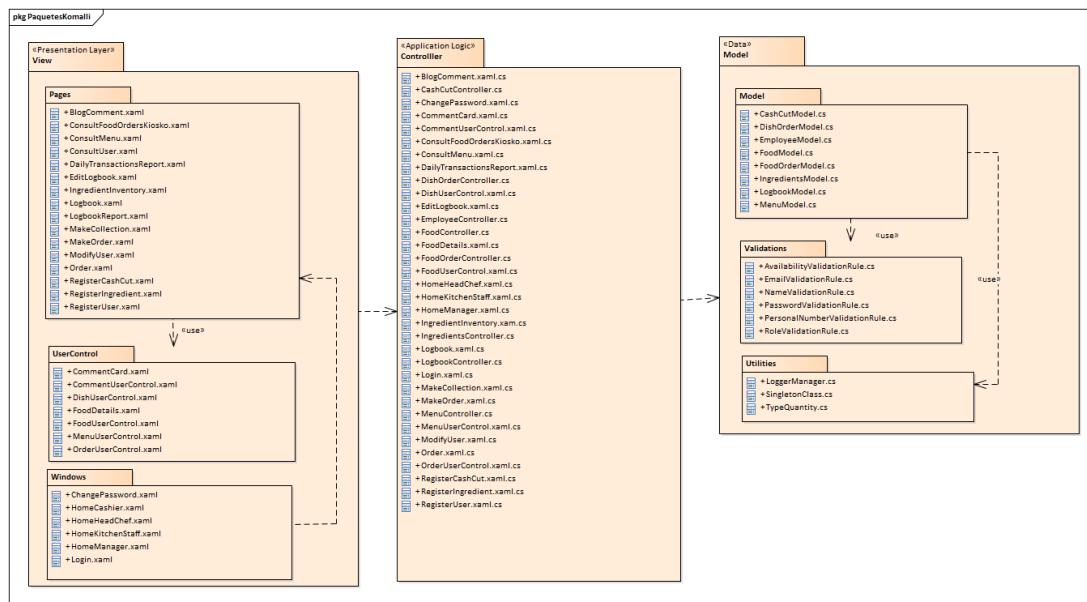


Ilustración 112 Diagrama de paquetes

3.28.2 El modelo de despliegue

El diagrama de despliegue muestra un sistema con dos dispositivos: una computadora cliente y una computadora servidor, ambos con Windows 11. En la computadora cliente se ejecutan las aplicaciones KomalliClient.sln y KomalliServer.sln. En la computadora servidor, se ejecuta SQL Server con la base de datos KomalliDB.sql la cual se aloja en un hosting llamado Somee. La comunicación entre el cliente y la base de datos en el servidor se realiza a través del protocolo ODBC.

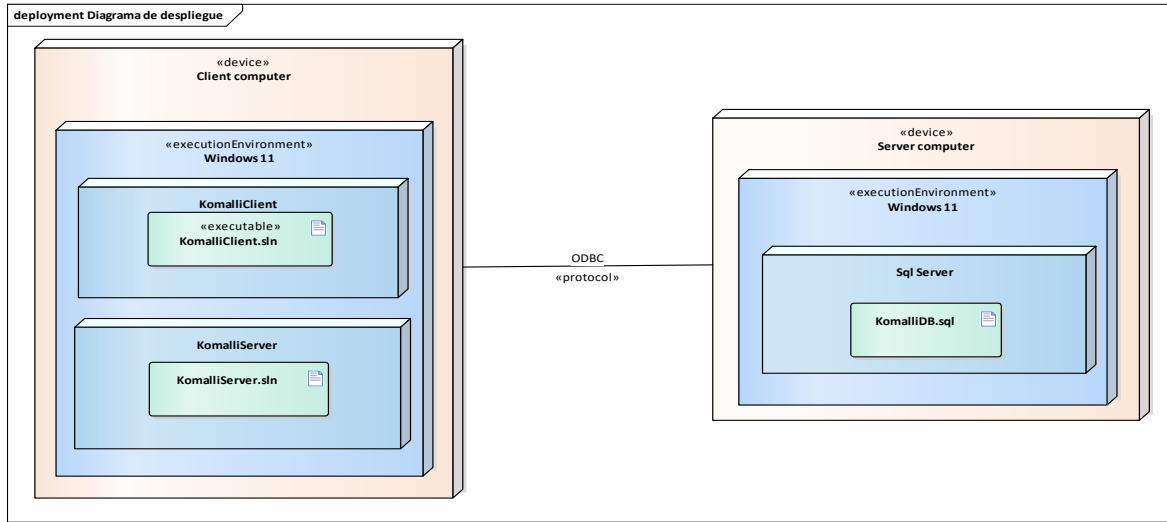


Ilustración 113 Diagrama de despliegue

3.28.3 Modelo de datos

Este modelo de datos gestiona pedidos de comida, empleados y registros financieros. Los usuarios (User) tienen información de acceso, mientras que los empleados (Employee) se asocian a usuarios y realizan pedidos (FoodOrder). Los pedidos incluyen menús (SetMenu) y cartas de menú (MenuCard). Los ingredientes (Ingredient) se registran con detalles de cantidad y reposición. Los cortes de caja (CashCut) gestionan el balance diario, y los registros (Logbook) documentan actividades de empleados. Las relaciones se manejan mediante tablas intermedias para conectar empleados con pedidos y pedidos con menús y cartas.

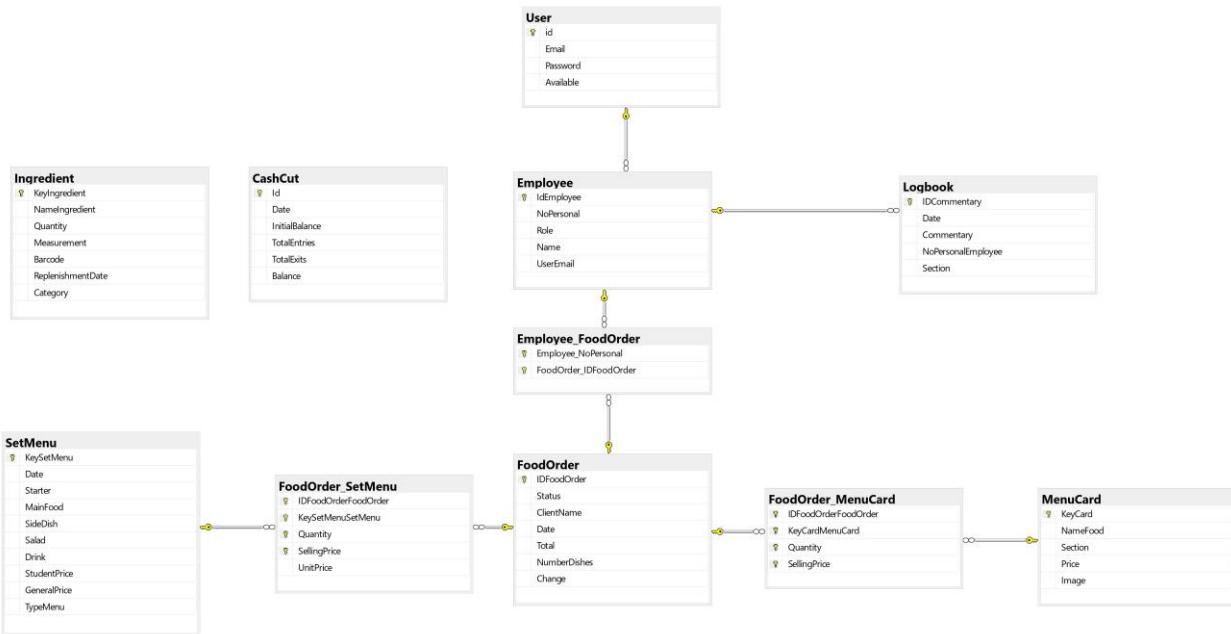


Ilustración 114 Modelo de datos

4 Pruebas

Las pruebas son una parte fundamental del desarrollo de software, asegurando que el sistema funcione correctamente y cumpla con los requisitos. Este apartado abarca tres tipos principales de pruebas: pruebas de unidad, que verifican el correcto funcionamiento de componentes individuales; pruebas de sistema, que aseguran que el sistema en su totalidad opere según lo esperado; y pruebas de integración, que validan la correcta interacción entre diferentes módulos y componentes del sistema.

4.1 Pruebas de unidad

Para la realización de las pruebas unitarias se utilizó la tecnología de MSTest. Se dividieron en dos paquetes: uno para las pruebas del caso del Cliente y otro para las del Empleado. Las pruebas se organizaron por módulos, y cada prueba unitaria se dividió en dos casos: uno exitoso, con la nomenclatura "Successful", y uno fallido, con la nomenclatura "Failed". Tanto las pruebas unitarias, como los modulos que contienen las mismas se nombraron de acuerdo al estándar definido por el equipo, el cual se encuentra en el apendice D.

A continuación, se enlistan las pruebas unitarias por módulo:

Paquete:

Client

Módulo: FoodController: este controlador maneja la obtención de alimentos por sección, la obtención de las claves de los menús del día, y el registro de pedidos tanto de platillos a la carta como de menús del día, manejando adecuadamente las excepciones y registrando errores cuando es necesario

- GetFoodsPerSection

▲ ✓ KomalliTest.Client (6)	7.6 s
▲ ✓ FoodControllerTest (2)	7.6 s
✓ GetFoodsPerSection_Failed	3.8 s
✓ GetFoodsPerSection_Sucessfull	3.8 s

Ilustración 115 Pruebas Modulo FoodController - Client

Modulo: MenuController: proporciona métodos para obtener los menús del día tanto para desayunos como para comidas.

- GetBreakfastOfTheDay
- GetFoodOfTheDay

▲ ✓	MenuControllerTest (4)	3.7 s
✓	GetBreakfastOfTheDay_Failed	3.5 s
✓	GetBreakfastOfTheDay_Sucessfull	50 ms
✓	GetFoodOfTheDay_Failed	44 ms
✓	GetFoodOfTheDay_Sucessfull	42 ms

Ilustración 116 Pruebas Modulo MenuController – Client

Paquete: Employee

Modulo: CashCutController: maneja la obtención del saldo inicial del último corte de caja, el registro de nuevos cortes de caja, y la actualización de cortes de caja existentes.

- GetLastInitialBalance
- RegisterCashCutNextDay
- UpdateDailyCashCut

▲ ✓	CashCutControllerTest (6)	740 ms
✓	GetLastInitialBalance_Failed	81 ms
✓	GetLastInitialBalance_Sucessfull	41 ms
✓	RegisterCashCutNextDay_Failed	117 ms
✓	RegisterCashCutNextDay_Suces...	151 ms
✓	UpdateDailyCashCut_Failed	192 ms
✓	UpdateDailyCashCut_Sucessfull	158 ms

Ilustración 117 Pruebas Modulo CashCutController - Employee

Modulo: DishOrderController: maneja la obtención de detalles de los pedidos tanto de los platillos de la carta como de los menús del día, proporcionando métodos específicos para cada tipo y combinando los resultados cuando se consultan todos los detalles de un pedido.

- ConsultDishOrderFromCard
- ConsultDishOrderFromMenu
- ConsultDishOrder

►	<input checked="" type="checkbox"/> DishOrderControllerTest (6)	3.9 s
	<input checked="" type="checkbox"/> ConsultDishOrder_Failed	97 ms
	<input checked="" type="checkbox"/> ConsultDishOrder_Sucesssfull	80 ms
	<input checked="" type="checkbox"/> ConsultDishOrderFromCard_Fai...	3.5 s
	<input checked="" type="checkbox"/> ConsultDishOrderFromCard_Su...	151 ms
	<input checked="" type="checkbox"/> ConsultDishOrderFromMenu_F...	61 ms
	<input checked="" type="checkbox"/> ConsultDishOrderFromMenu_S...	41 ms

Ilustración 118 Pruebas Modulo DishOrderController - Employee

Modulo: EmployeeController: El controlador gestiona operaciones relacionadas con empleados y usuarios en la base de datos de la aplicación. Permite obtener y actualizar roles y contraseñas de usuarios, validar usuarios, agregar nuevos usuarios y empleados, y verificar la existencia de correos electrónicos y números de personal. Además, proporciona métodos para obtener información detallada de los usuarios y empleados, y para transformar valores de disponibilidad entre representaciones numéricas y de texto. Maneja excepciones y registra errores usando un sistema de logging.

- AddEmployee
- AddUser
- ConsultUsers
- GetNoPersonalEmployee
- GetUserInfo
- GetUserName
- GetUserRole
- UpdateEmployeeInfo
- UpdatePassword
- UpdateUserInfo
- ValidateEmail
- ValidatePersonalNumber
- ValidateUser

EmployeeControllerTest (26)	36.9 s
AddEmployee_Failed	3.4 s
AddEmployee_Sucessfull	270 ms
AddUser_Failed	3.6 s
AddUser_Sucessfull	249 ms
ConsultUsers_Failed	3.3 s
ConsultUsers_Sucessfull	40 ms
GetNoPersonalEmployee_Failed	39 ms
GetNoPersonalEmployee_Suces...	40 ms
GetUserInfo_Failed	40 ms
GetUserInfo_Sucessfull	51 ms
GetUserName_Failed	41 ms
GetUserName_Sucessfull	3.8 s
GetUserRule_Failed	117 ms
GetUserRule_Sucessfull	3.8 s
UpdateEmployeeInfo_Failed	116 ms
UpdateEmployeeInfo_Sucessfull	169 ms
UpdatePassword_Failed	255 ms
UpdatePassword_Sucessfull	4.9 s
UpdateUserInfo_Failed	160 ms
UpdateUserInfo_Sucessfull	169 ms
ValidateEmail_Failed	4.4 s
ValidateEmail_Sucessfull	42 ms
ValidatePersonalNumber_Failed	3.4 s
ValidatePersonalNumber_Suces...	45 ms
ValidateUser_Failed	4.3 s
ValidateUser_Sucessfull	213 ms

Ilustración 119 Pruebas Modulo EmployeeController - Employee

Modulo: FoodController: es responsable de gestionar varias operaciones relacionadas con los alimentos y los pedidos dentro del sistema Komalli. Esta clase incluye métodos para recuperar alimentos por sección, obtener claves de menú diarias y registrar pedidos de alimentos.

- GetFoodsPerSection
- GetKeyMenu
- GetStockMenuCard
- GetStockSetMenu
- RegistryOrderMenu
- RegistryOrderMenuCard
- UpdateStockMenuCard
- UpdateStockSetMenu

	FoodControllerTest (16)	5.1 s
✓	GetFoodsPerSection_Failed	45 ms
✓	GetFoodsPerSection_Sucessfull	58 ms
✓	GetKeyMenu_Failed	58 ms
✓	GetKeyMenu_Sucessfull	41 ms
✓	GetStockMenuCard_Failed	97 ms
✓	GetStockMenuCard_Sucessfull	46 ms
✓	GetStockSetMenu_Failed	42 ms
✓	GetStockSetMenu_Sucessfull	46 ms
✓	RegistryOrderMenu_Failed	181 ms
✗	RegistryOrderMenu_Sucessfull	3.4 s
✓	RegistryOrderMenuCard_Failed	146 ms
✓	RegistryOrderMenuCard_Suces...	325 ms
✓	UpdateStockMenuCard_Failed	168 ms
✓	UpdateStockMenuCard_Sucessf...	186 ms
✓	UpdateStockSetMenu_Failed	131 ms
✓	UpdateStockSetMenu_Sucessfull	173 ms

Ilustración 120 Pruebas Modulo FoodController - Employee

Modulo: FoodOrderController: Este controlador se encarga de la gestión de las ordenes hechas por los clientes. Este recibirá las ordenes realizadas por el kioko o se encargará de registrar las ordenes

hechas por los clientes desde caja. De igual manera, se podrá actualizar o eliminar las ordenes, al actualizar se refiere de cambiar el estado de un pedido. También, calculará las entradas y salidas hechas en un día y podrá obtener todos los pedidos realizados en determinado día.

- CalculateTotalDailyChange
- CalculateTotalDailyEntries
- ConsultFoodOrdersKiosko
- DailyFoodOrders
- DeleteOrder
- GetTotalAndNameFromOrder
- RegistryOrder
- UpdateFoodOrder
- UpdateOrderStatus

▲	<input checked="" type="checkbox"/> FoodOrderControllerTest (18)	12.8 s
	<input checked="" type="checkbox"/> CalculateTotalDailyChange_Failed	48 ms
	<input checked="" type="checkbox"/> CalculateTotalDailyChange_Suc...	206 ms
	<input checked="" type="checkbox"/> CalculateTotalDailyEntries_Failed	41 ms
	<input checked="" type="checkbox"/> CalculateTotalDailyEntries_Suce...	72 ms
	<input checked="" type="checkbox"/> ConsultFoodOrdersKiosko_Failed	3.7 s
	<input checked="" type="checkbox"/> ConsultFoodOrdersKiosko_Suce...	3.7 s
	<input checked="" type="checkbox"/> DailyFoodOrders_Failed	41 ms
	<input checked="" type="checkbox"/> DailyFoodOrders_Sucessfull	3.6 s
	<input checked="" type="checkbox"/> DeleteOrder_Failed	124 ms
	<input checked="" type="checkbox"/> DeleteOrder_Sucessfull	157 ms
	<input checked="" type="checkbox"/> GetTotalAndNameFromOrder_F...	42 ms
	<input checked="" type="checkbox"/> GetTotalAndNameFromOrder_S...	65 ms
	<input checked="" type="checkbox"/> RegistryOrder_Failed	176 ms
	<input checked="" type="checkbox"/> RegistryOrder_Sucessfull	126 ms
	<input checked="" type="checkbox"/> UpdateFoodOrder_Failed	186 ms
	<input checked="" type="checkbox"/> UpdateFoodOrder_Sucessfull	168 ms
	<input checked="" type="checkbox"/> UpdateOrderStatus_Failed	184 ms
	<input checked="" type="checkbox"/> UpdateOrderStatus_Sucessfull	203 ms

Ilustración 121 Pruebas Modulo FoodOrderController - Employee

<input checked="" type="checkbox"/> UpdateOrderStatus_Failed	144 ms
<input checked="" type="checkbox"/> UpdateOrderStatus_Sucessfull	517 ms

Ilustración 122 Pruebas Modulo FoodOrderController - Employee

Modulo: IngredientController: El controlador de ingredientes maneja las operaciones CRUD y de búsqueda para los ingredientes en la base de datos.

- AddIngredient
- IsBarcodeExisting
- IsNameIngredientExisting
- ModifyIngredients
- SearchIngredients

►	✓ IngredientControllerTest (10)	1.6 s
	✓ AddIngredient_Failed	136 ms
	✓ AddIngredient_Sucessfull	127 ms
	✓ IsBarcodeExisting_Failed	140 ms
	✓ IsBarcodeExisting_Sucessfull	170 ms
	✓ IsNameIngredientExisting_Failed	129 ms
	✓ IsNameIngredientExisting_Suce...	145 ms
	✓ ModifyIngredients_Failed	211 ms
	✓ ModifyIngredients_Sucessfull	210 ms
	✓ SearchIngredients_Failed	143 ms
	✓ SearchIngredients_Sucessfull	158 ms

Ilustración 123 Pruebas IngredientController - Employee

Modulo: LogbookController: El controlador maneja la gestión de los comentarios en la bitácora, permitiendo operaciones como añadir, obtener, actualizar y eliminar comentarios.

- AddComment
- DeleteComment
- GetEmployeeComments
- UpdateComment

►	✗ LogbookControllerTest (8)	26.5 s
	✗ AddComment_Failed	181 ms
	✓ AddComment_Sucessfull	3.7 s
	✗ DeleteCommentary_Failed	3.7 s
	✗ DeleteCommentary_Sucessfull	3.7 s
	✓ GetEmployeeComments_Failed	3.7 s
	✓ GetEmployeeComments_Sucess...	4 s
	✓ UpdateComment_Failed	3.8 s
	✓ UpdateComment_Sucessfull	3.7 s

Ilustración 124 Pruebas Modulo LogbookController - Employee

Modulo: SetMenuController: El controlador gestiona los menús en la base de datos, permitiendo

operaciones como agregar, eliminar, verificar la existencia, y modificar menús. Además, maneja la verificación de menús existentes en base a la fecha y tipo.

- AddSetMenu
- DeleteSetMenu
- ExistingMenu
- ExistingTypeMenu
- ModifySetMenu
- UpdateMenu

▲ ✓ SetMenuControllerTest (10)	5.6 s
✓ AddSetMenu_Failed	305 ms
✓ AddSetMenu_Sucessfull	124 ms
✓ DeleteSetMenu_Failed	77 ms
✓ DeleteSetMenu_Sucessfull	44 ms
✓ ExistingMenu_Sucessfull	148 ms
✓ ExistingTypeMenu_Failed	143 ms
✓ ExistingTypeMenu_Sucessfull	58 ms
✓ ModifySetMenu_Failed	187 ms
✓ ModifySetMenu_Sucessfull	221 ms
✓ UpdateMenu_Sucessfull	4.3 s

Ilustración 125 Pruebas Modulo SetMenuController - Employee

Como resultado, se lograron pasar un total de 110 pruebas unitarias. De las cuales 4 no pasaron debido a la gran cantidad de registros.

The screenshot shows the JUnit test results interface. At the top, there are various icons for navigating through the tests. Below that, a summary message reads: "Serie de pruebas finalizada: 1 pruebas (Superadas: 0; Con errores: 1; Omitidas: 0) ejecutadas en 5.6 s". A table follows, with columns for "Prueba", "Duración", "Rasgos", and "Mensaje de error". The table data is as follows:

Prueba	Duración	Rasgos	Mensaje de error
▲ ✗ KomalliTest (110)	58.2 s		
✓ KomalliTest.Client (6)	317 ms		
✗ KomalliTest.Employee (104)	57.9 s		

Ilustración 126 Pruebas totales

4.2 Pruebas de integración

Las pruebas de integración son esenciales para asegurar que los diferentes módulos de un sistema funcionen correctamente al combinarse, verificando que los flujos principales operen sin problemas y que la interacción entre los casos de uso sea adecuada. En nuestro escenario, estas pruebas confirmaron que los flujos principales del sistema funcionaron de manera efectiva con todos los usuarios involucrados. A continuación, en la Tabla 27, se detallan las pruebas realizadas así como el resultado obtenido en cada una de ellas y posteriormente se muestran las evidencias de que las pruebas fueron realizadas correctamente.

# C P	Flujo	#CU involucrados	Entrada	R. obtenido	R. esperado	Estado
1	Empleado con rol de Cajero, realiza un pedido, cobra el mismo. Un empleado con rol de Personal de cocina consulta los pedidos, y modifica su estado.	#CU03. Levantar pedido	1 club sándwich 1 huevo revuelto	Registro exitoso	Registro exitoso	Verde
		#CU20. Realizar cobro	Recibe: \$100 Nombre: Alejandro Marín	Registro exitoso	Registro exitoso	Verde
		#CU11. Editar estado de pedido	Cambiar a estado de "Hecho"	Cambio de estado exitoso	Cambio de estado exitoso	Verde
		#CU11. Editar estado de pedido	Cambiar a estado de "Entregado "	Cambio de estado exitoso	Cambio de estado exitoso	Verde
2	Empleado con rol de jefe de cocina, registra un ingrediente en el	#CU05. Registrar ingrediente en inventario	Nombre: Papas Cantidad: 30 Categoría: Frutas y verduras	El ingrediente fue agregado correctamente	El ingrediente fue agregado correctamente	Verde

	inventario. El empleado agrega un comentario en la bitácora en la sección de merma (si es que sucede alguna incidencia con el inventario). El gerente realiza reporte de inventario y bitácora		Medición: 30 unidades			
	#CU16. Agregar comentario a bitácora	Comentario: Se rompieron 3 platos hondos. Categoría: Incidencia	El comentario se ha publicado correctamente	El comentario se ha publicado correctamente		
	#CU08. Generar reporte de inventario	Filtrado por: Frutas y verduras	Reporte descargado correctamente	Reporte descargado correctamente		
	#CU19. Generar PDF de la bitácora	Filtrar por: Incidencias Filtrar por: 22/07/2024	Reporte descargado correctamente	Reporte descargado correctamente		
3	Empleado con rol de cajero, hace corte de caja. El gerente visualiza los movimientos	#CU22. Realiza corte de caja	Saldo inicial: \$500	Corte de caja registrado exitosamente	Corte de caja registrado exitosamente	

	diarios y de manera opcional, descarga el reporte.	#CU23. Generar PDF de movimientos diarios	Fecha inicio: 16/05/2024 Fecha fin: 16/05/2022	Reporte descargado correctamente	Reporte descargado correctamente	
--	--	---	---	----------------------------------	----------------------------------	--

Tabla 27. Pruebas de integración



Ilustración 127 CP.01 Empleado ingresa al sistema a hacer un pedido



Ilustración 128 CP.01 Empleado realiza el cobro de un pedido



Ilustración 129 CP.01 Empleado consulta pedidos

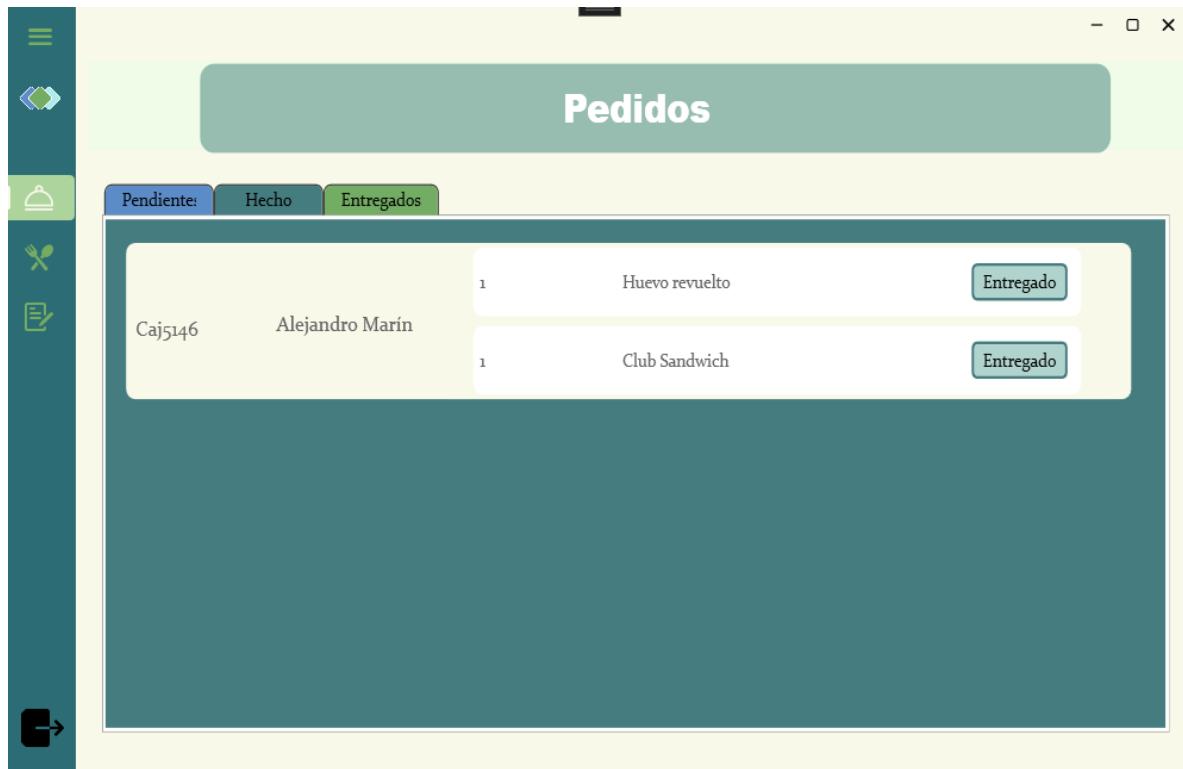


Ilustración 130 CP.01 Empleado modifica pedido a "Hecho"



Ilustración 131 CP.01 Empleado modifica pedido a "Entregado"

Registro de ingredientes

Nombre del ingrediente
Papas

Cantidad
30

Categoría
Frutas y Verduras

Medición
Unidades

Código de barras (Opcional)

Registrar Ingrediente

Ilustración 132 CP.02 Empleado registra ingrediente

Clave	Categoría	Nombre	Cantidad	Medida	Actualización
PAPA3516	Frutas y Verduras	Papas	30	Unidades	22/05/2024

Ilustración 133 CP.02 Empleado consulta ingrediente registrado

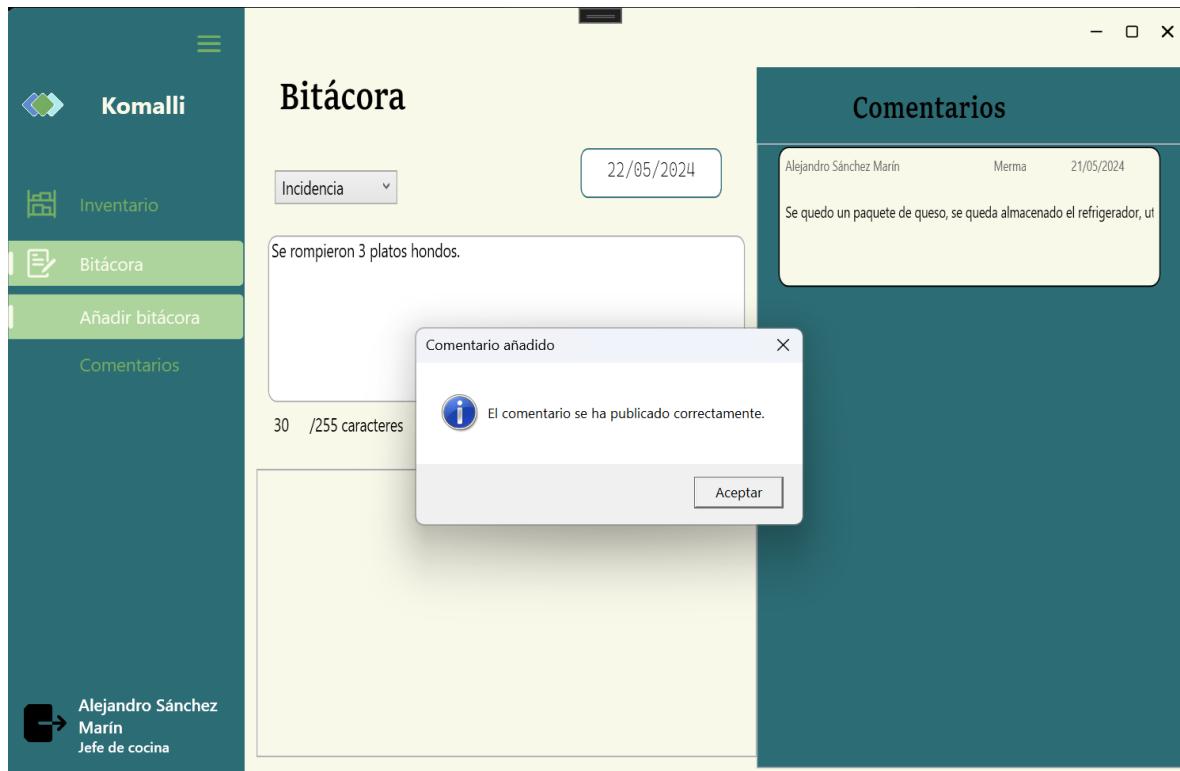


Ilustración 134 CP.02 Empleado agrega comentario a bitácora

The screenshot shows the Komalli software interface. On the left, there's a sidebar with icons for Inventory, Log, Add Log, and Comments. The main area is titled "Reporte de inventario" and shows a filter dropdown set to "Frutasyverduras" and a "Descargar" button. A modal window titled "Reporte de inventario" displays a table of inventory items:

Clave	Nombre	Cantidad	Medición	Categoría
CILA5565	Cilantro	10	Unidades	FrutasyVerduras
CILA7301	Perejil	15	Unidades	FrutasyVerduras
FRESS5988	Fresas	10	Kg	FrutasyVerduras
JITO5262	Jitomate	10	Kg	FrutasyVerduras
LECH4396	Lechuga	10	Kg	FrutasyVerduras
LECH9564	Lechuga Romana	.5	Unidades	FrutasyVerduras
MERM8100	Mermelada de fresa	.3	Kg	FrutasyVerduras
PAPA3516	Papas	30	Unidades	FrutasyVerduras
PIÑA7024	Piña	5	Kg	FrutasyVerduras

Ilustración 135 CP.02 Gerente consulta reporte de inventario

Reporte de inventario

Clave	Nombre	Cantidad	Medición	Categoría
CILA5565	Cilantro	10	Unidades	Frutas y Verduras
CILA7301	Perejil	15	Unidades	Frutas y Verduras
FRES5988	Fresas	10	Kg	Frutas y Verduras
JITO5262	Jitomate	10	Kg	Frutas y Verduras
LECH4396	Lechuga	10	Kg	Frutas y Verduras
LECH9564	Lechuga Romana	.5	Unidades	Frutas y Verduras
MERM8100	Mermelada de fresa	.3	Kg	Frutas y Verduras
PAPA3516	Papas	30	Unidades	Frutas y Verduras
PIÑA7024	Piña	.5	Kg	Frutas y Verduras

Ilustración 136 CP.02 Gerente descarga reporte de inventario

Reporte de bitácora

Filtrar por: 22/05/2024 Descargar

Filtrar por: Incidencias

Nombre	Comentario	Fecha	Rol	Sección
Ares Juda Rivera Soto	Se han comprado tres paquetes de servilletas	22/05/2024	Cajero	General
Ares Juda Rivera Soto	Se han caducado dos paquetes de jugos	22/05/2024	Cajero	Merma
Ares Juda Rivera Soto	Presentarse a una reunión informal el día viernes 25 de mayo 10 minutos antes de la hora de entrada	22/05/2024	Cajero	Otro
Ares Juda Rivera Soto	Hoy sobraron 5 menús de comida de los chicos becados	22/05/2024	Cajero	General
Ares Juda Rivera Soto	El día viernes se tiene programado hacer crema de zanahoria, necesitaremos más mantequilla	22/05/2024	Cajero	General
Camila	Un alumno tiró los platos limpios por accidente	22/05/2024	Jefe de cocina	General
Alejandro Sánchez Marín	Se rompieron 3 platos hondos.	22/05/2024	Jefe de cocina	Incidencia

Ilustración 137 CP.02 Gerente consulta reporte de bitácora

Reporte de bitácora

Nombre	Comentario	Fecha	Rol	Sección
Ares Juda Rivera Soto	Se han comprado tres paquetes de servilletas	22/05/2024	Cajero	General
Ares Juda Rivera Soto	Se han caducado dos paquetes de jugos	22/05/2024	Cajero	Merma
Ares Juda Rivera Soto	Presentarse a una reunión informal el día viernes 25 de mayo 10 minutos antes de la hora de entrada	22/05/2024	Cajero	Otro
Ares Juda Rivera Soto	Hoy sobraron 5 menús de comida de los chicos becados	22/05/2024	Cajero	General
Ares Juda Rivera Soto	El día viernes se tiene programado hacer crema de zanahoria, necesitaremos más mantequilla	22/05/2024	Cajero	General
Camila	Un alumno tiró los platos limpios por accidente	22/05/2024	Jefe de cocina	General
Alejandro Sánchez Marín	Se rompieron 3 platos hondos.	22/05/2024	Jefe de cocina	Incidencia

Ilustración 138 CP.02 Gerente descarga reporte de bitácora

Corte de caja

Saldo inicial: 1000	Total de entradas: 530	Total de salidas: 167	Balance: 1363
-------------------------------	----------------------------------	---------------------------------	-------------------------

Fecha de corte: 16/05/2024 Saldo inicial para el siguiente día: 500

Realizar corte

Historial de ventas

N. Operación	Fecha	Nombre del cliente	N. de platillos	Subtotal
Caj0915	16/05/2024	pablo	3	42
Caj3731	16/05/2024	ysela lopez	3	95

Ilustración 139 CP.03 Empleado consulta corte de caja

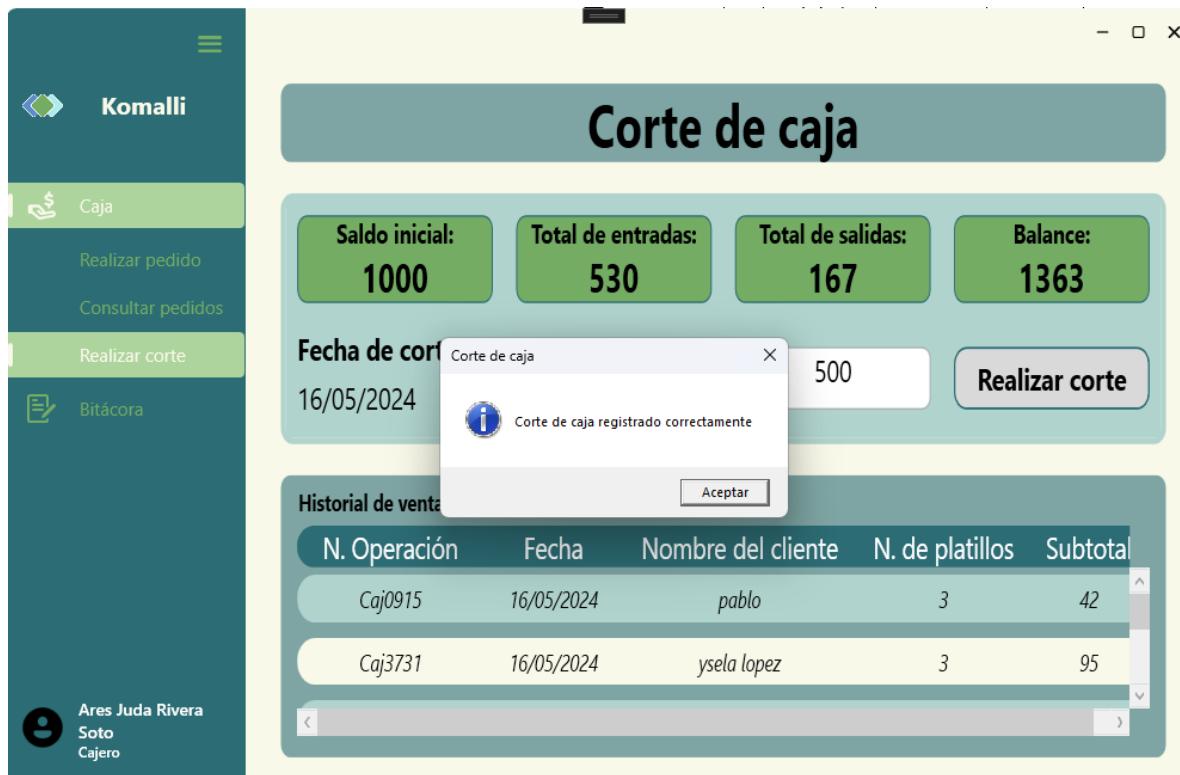


Ilustración 140 CP.03 Empleado realiza corte de caja



Ilustración 141 CP.03 Gerente consulta reporte de movimientos diarios

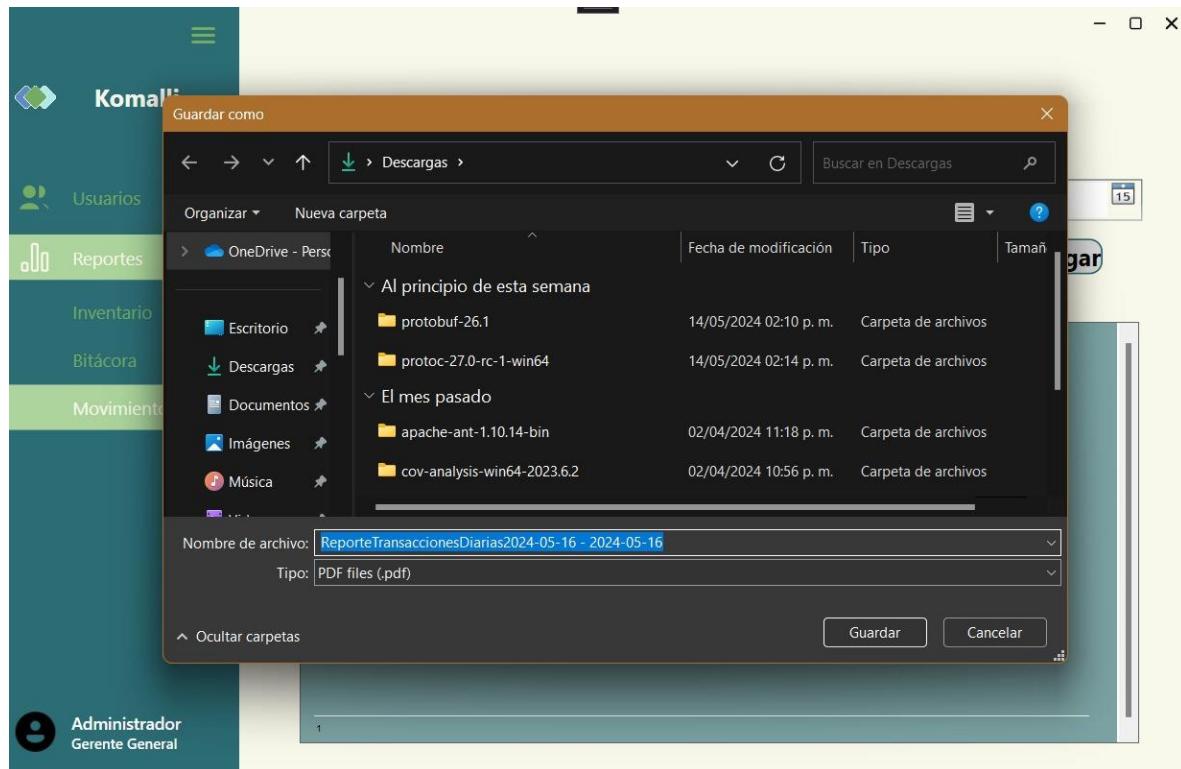


Ilustración 142 CP.03 Gerente descarga reporte de movimientos diarios

N. Operación	Fecha	Empleado	N. platillos	Subtotal
Caj0915	16/05/2024	Ysela Lopez	3	\$42
Caj731	16/05/2024	Jesus Suarez	3	\$95
Caj6460	16/05/2024	Carlos Hdez	6	\$27
Caj9634	16/05/2024	Paloma	1	\$288
Caj9957	16/05/2024			\$48

Ilustración 143 CP.03 Descarga exitosa del reporte

4.3 Pruebas de sistema

Las pruebas de sistema son cruciales para garantizar que una aplicación funcione correctamente en su totalidad y cumpla con los requisitos especificados. Un aspecto fundamental de estas pruebas es verificar que los usuarios, al ingresar con ciertos roles, tengan acceso únicamente a las funcionalidades asociadas a esos roles. Esto asegura que cada usuario interactúe con el sistema de manera segura y conforme a sus permisos, manteniendo la integridad y seguridad de la aplicación. A continuación, en la Tabla 28, se detallan las pruebas realizadas así como el resultado obtenido en cada una de ellas y posteriormente se muestran las evidencias de que las pruebas fueron realizadas correctamente.

# C P	Flujo	#CU involucrados	Entrada	R. obtenido	R. esperado	Estado
1	Registrar nuevo empleado con rol de cajero. Validar que solo tenga acceso a las secciones de Caja y bitácora	#CU01. Registrar usuario Login	Nombre: <i>Diego Cardone</i> No.Personal: 70777 Correo: diego@gmail.com Rol: <i>Cajero</i>	Registro exitoso	Registro exitoso	
2	Registrar nuevo empleado con rol de Jefe de cocina. Validar que solo tenga acceso a las secciones de inventario y bitácora	#CU01. Registrar usuario Login	Nombre: <i>Juan López Grajales</i> No.Personal: 28415 Correo: djuanL@gmail.com Rol: <i>Jefe de cocina</i>	Registro exitoso	Registro exitoso	

3	Registrar nuevo empleado con rol de Personal de cocina. Validar que solo tenga acceso a las secciones de Menú, pedidos y bitácora	#CU01. Registrar usuario Login	Nombre: <i>Janet García Mora</i> No.Personal: 35014 Correo: janet@gmail.com Rol: Personal de cocina	Registro exitoso	Registro exitoso	
4	Acceder al sistema como Gerente y validar que solo tenga acceso a las secciones de Usuarios y Reportes	Login	No aplica	Login exitoso	Login exitoso	

Tabla 28. Pruebas de sistema



Ilustración 144 CP.01 Registrar cajero



Ilustración 145 CP.01 Iniciar sesión como cajero



Ilustración 146 CP.02 Registrar jefe de cocina

Inventario de ingredientes

Clave	Nombre	Cantidad	Medida	Actualización
.5866	Mermelada de Fré	2	Unidades	
AGUA1653	Aguacate	50	Kg	
AGUA8699	Agua natural 500	10	Unidades	
CILA5565	Cilantro	10	Unidades	
CILA7301	Perejil	15	Unidades	
GALL9888	Galletas de choco	15	Unidades	

Ilustración 147 CP.02 Iniciar sesión como jefe de cocina



Ilustración 148 CP.03 Registrar personal de cocina

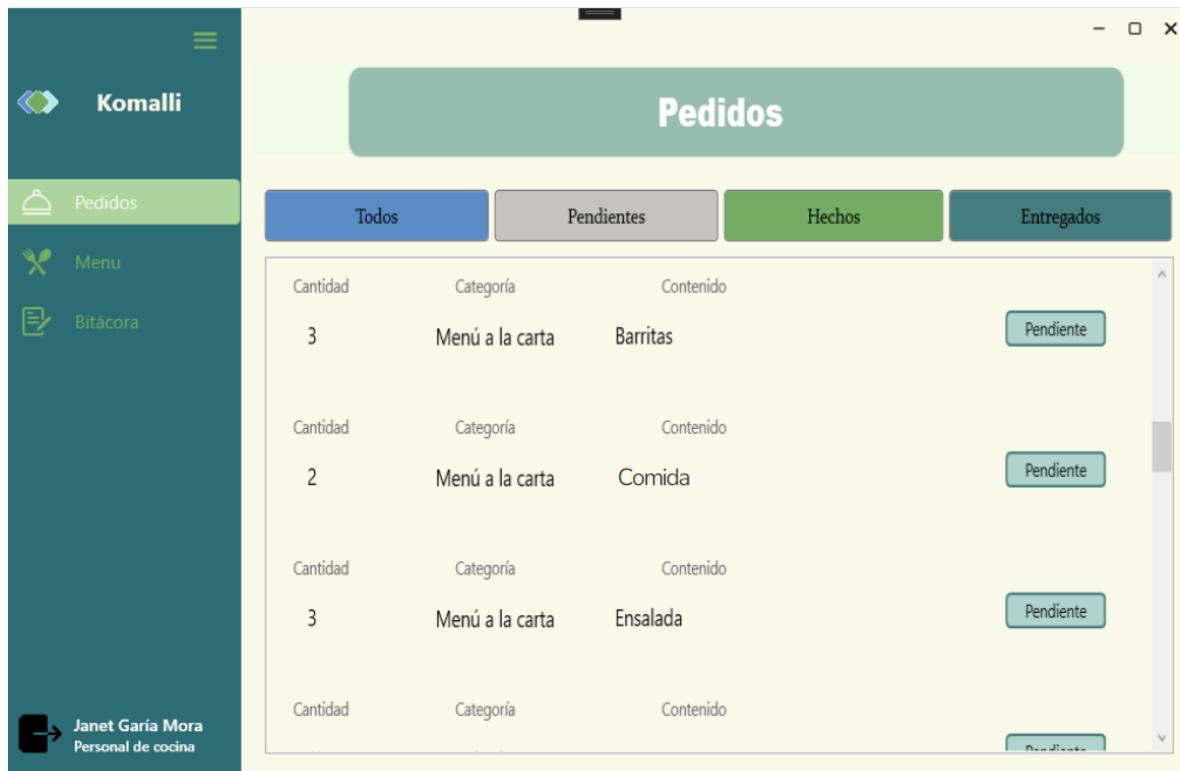


Ilustración 149 CP.03 Iniciar sesión como personal de cocina

The screenshot shows the Komalli software interface. On the left is a sidebar with the Komalli logo and a navigation menu:

- Usuarios
- Consultar
- Registrar
- Reportes
- Inventario
- Bitácora
- Movimientos diarios

At the bottom of the sidebar is a button for "Administrador Gerente General".

The main area is titled "Lista de usuarios registrados" (List of registered users). It features a search bar labeled "Buscar usuario" with a magnifying glass icon. Below the search bar is a table with the following columns: No. Personal, Nombre, Rol, and Estatus.

No. Personal	Nombre	Rol	Estatus
111	Ares Juda Rivera Soto	Cajero	Activo
2910	Gerardo Sánchez Marín	Personal de cocina	Activo
628	Ysela Lara Landa	Cajero	Inactivo
777	Alejandro Sánchez Marín	Jefe de cocina	Activo
70777	Diego Cardone	Cajero	Activo
28410	Juan Lopez Gutierrez	Jefe de cocina	Activo
28415	Juan López Grajales	Jefe de cocina	Activo
35014	Janet García Mora	Personal de cocina	Activo

Ilustración 150 CP.04 Iniciar sesión como gerente

Conclusiones

Logros, problemas, soluciones, trabajos futuros

A lo largo de estas 13 semanas de arduo trabajo, hemos alcanzado nuestra meta principal: la correcta finalización del sistema de administración para el comedor universitario Komalli. Nos enorgullece destacar que logramos completar este proyecto tres semanas antes de la fecha de entrega. Este logro se debe a la dedicación y compromiso de todos los integrantes del equipo.

Una de las principales características de nuestro equipo fue la comunicación efectiva y la identificación temprana de riesgos, lo que nos permitió gestionar el proyecto y el trabajo de manera óptima. Gracias a estos factores, pudimos cumplir nuestra meta antes del tiempo previsto.

Sin embargo, también enfrentamos problemas durante el desarrollo. Por ejemplo, un error en un merge del repositorio nos llevó a crear una nueva rama para resolver los conflictos de código que surgieron. Además, la caducidad de nuestros créditos en Azure nos obligó repetidamente a cambiar la ubicación y respaldo de nuestra base de datos en la nube. Ante esto, tuvimos que buscar alternativas, y Somme resultó ser la opción más viable para el proyecto.

Para futuros trabajos, consideramos seguir con este tipo de organización y métricas de trabajo. Creemos que uno de los puntos más acertados en el desarrollo del proyecto fue la creación temprana de todos los prototipos, lo que nos permitió tener desde el comienzo una visión general del sistema. Esto facilitó la identificación de casos de uso, tanto faltantes como sobrantes, y es una práctica que planeamos adoptar en futuros proyectos.

La constante comunicación dentro del equipo también fue crucial para nuestro éxito. Además, consideramos que realizar pruebas inmediatamente después de finalizar cada caso de uso contribuyó significativamente al éxito del proyecto. Al hacer las pruebas de inmediato, evitamos acumular una gran deuda técnica que podría haber afectado el proyecto más adelante.

Bibliografía

- Fakhroutdinov, K. Unified Modeling Language (UML) description, UML diagram examples, tutorials and reference for all types of UML diagrams - use case diagrams, class, package, component, composite structure diagrams, deployments, activities, interactions, profiles, etc. Recuperado el 27 de mayo de 2024, de <https://www.uml-diagrams.org/>
- de Josué Andrade Mirós Octubre de, J. M. F. P. y. M. de L. Á. S. L. C. (s/f). Notas del método con ampliaciones y mejoras. Www.uv.mx. Recuperado el 28 de mayo de 2024, de <https://www.uv.mx/personal/asumano/files/2010/07/iconix2.pdf>
- Prueba unitaria de C# con MSTest y .NET. (s/f). Microsoft.com. Recuperado el 27 de mayo de 2024, de <https://learn.microsoft.com/es-es/dotnet/core/testing/unit-testing-with-mstest>
- Jeyderht. C# WPF - UI Design || Menu Vertical (Demo) [Vídeo]. YouTube. Recuperado el 27 de mayo de 2024, de <https://www.youtube.com/watch?v=IhhDDQLYKPE>

Apéndice A Diagrama de clases

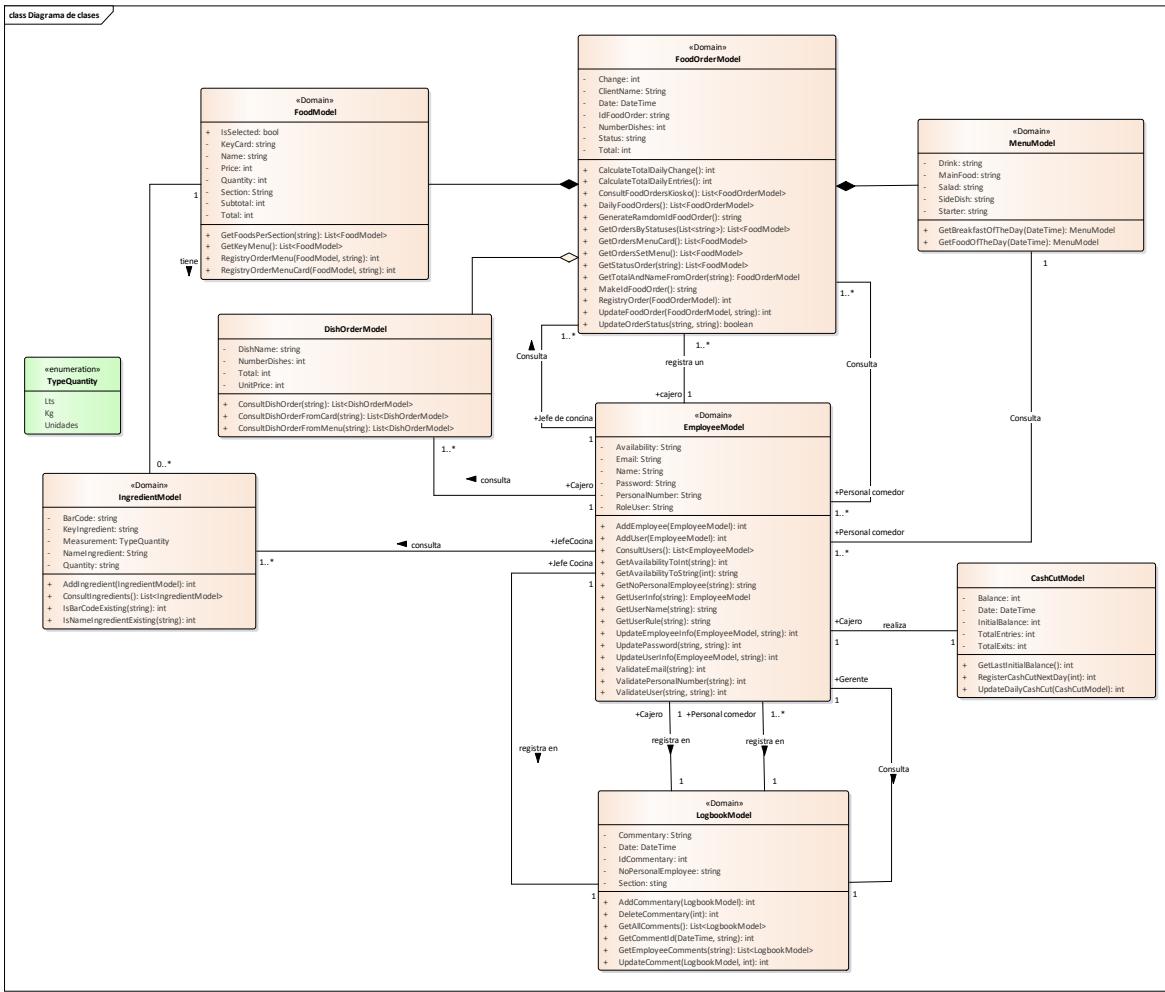


Ilustración 151 Diagrama de clases

Apéndice B Diccionario de Datos

Tabla: CashCut

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
Id	int	N/A	No	Identificador único del corte de caja.
Date	date	N/A	Sí	Fecha del corte de caja.
InitialBalance	int	N/A	Sí	Saldo inicial al comenzar el día.
TotalEntries	int	N/A	Sí	Total de entradas de efectivo del día.
TotalExits	int	N/A	Sí	Total de salidas de efectivo del día.
Balance	int	N/A	Sí	Balance final del día (InitialBalance + TotalEntries - TotalExits).

Tabla 29. Diccionario de datos - CashCut

Tabla: Employee

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
IdEmployee	int	N/A	No	Identificador único del empleado.
NoPersonal	varchar	5	No	Número personal del empleado, único para cada empleado.
Role	varchar	20	No	Rol del empleado (e.g., Gerente, cajero, personal de cocina, etc.).
Name	varchar	30	No	Nombre completo del empleado.
UserEmail	varchar	255	No	Correo electrónico del usuario asociado al empleado.

Tabla 30. Diccionario de datos - Employee

Tabla: Employee_FoodOrder

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
Employee_NoPersonal	varchar	5	No	Número personal del empleado que realiza el pedido.
FoodOrder_IDFoodOrder	varchar	10	No	Identificador del pedido de comida realizado por el empleado.

Tabla 31. Diccionario de datos - Employee_FoodOrder

Tabla: FoodOrder

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
IDFoodOrder	varchar	10	No	Identificador único del pedido de comida.
Status	varchar	10	No	Estado del pedido (e.g., pendiente, completado, cancelado).
ClientName	varchar	50	Sí	Nombre del cliente que realiza el pedido.

Date	date	N/A	No	Fecha en que se realizó el pedido.
Total	int	N/A	No	Monto total del pedido.
NumberDishes	int	N/A	No	Número total de platos en el pedido.
Change	int	N/A	Sí	Cambio devuelto al cliente, si aplica.

Tabla 32. Diccionario de datos - FoodOrder

Tabla: FoodOrder_MenuCard

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
IDFoodOrderFoodOrder	varchar	10	No	Identificador del pedido de comida asociado.
KeyCardMenuCard	varchar	10	No	Clave del elemento del menú ordenado.
Quantity	int	N/A	No	Cantidad del elemento del menú ordenado.
SellingPrice	int	N/A	No	Precio de venta del elemento del menú.

Tabla 33. Diccionario de datos - FoodOrder_MenuCard

Tabla: FoodOrder_SetMenu

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
IDFoodOrderFoodOrder	varchar	10	No	Identificador del pedido de comida asociado.
KeySetMenuItem	varchar	10	No	Clave del menú fijo ordenado.
Quantity	int	N/A	No	Cantidad del menú fijo ordenado.
SellingPrice	int	N/A	No	Precio de venta del menú fijo.
UnitPrice	int	N/A	Sí	Precio unitario del menú fijo, si aplica.

Tabla 34. Diccionario de datos - FoodOrder_SetMenu

Tabla: Ingredient

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
KeyIngredient	varchar	10	No	Clave única del ingrediente.
NameIngredient	varchar	30	No	Nombre del ingrediente.
Quantity	varchar	10	No	Cantidad disponible del ingrediente.
Measurement	varchar	10	No	Unidad de medida de la cantidad del ingrediente.
Barcode	varchar	13	Sí	Código de barras del ingrediente, si aplica.
ReplenishmentDate	date	N/A	Sí	Fecha de reposición del ingrediente.
Category	varchar	55	No	Categoría a la que pertenece el ingrediente (e.g., lácteos, verduras).

Tabla 35. Diccionario de datos - Ingredient

Tabla: Logbook

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
IDCommentary	int	N/A	No	Identificador único del comentario en el libro de registro.
Date	date	N/A	No	Fecha en que se realizó el comentario.
Commentary	varchar	255	No	Texto del comentario realizado.
NoPersonalEmployee	varchar	5	No	Número personal del empleado que realizó el comentario.
Section	varchar	55	No	Sección a la que se refiere el comentario (e.g., cocina, caja, servicio).

Tabla 36. Diccionario de datos - Logbook

Tabla: MenuCard

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
KeyCard	varchar	10	No	Clave única del elemento en la carta del menú.
NameFood	varchar	25	No	Nombre del alimento en la carta del menú.
Section	varchar	15	No	Sección del menú a la que pertenece el alimento (e.g., entradas, platos fuertes, postres).
Price	int	N/A	No	Precio del alimento en la carta del menú.
Image	varchar	100	Sí	Ruta de la imagen del alimento del menú.

Tabla 37. Diccionario de datos - MenuCard

Tabla: SetMenu

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
KeySetMenu	varchar	10	No	Clave única del menú fijo.
Date	date	N/A	No	Fecha en que se ofrece el menú fijo.
Starter	varchar	20	Sí	Nombre del plato de entrada incluido en el menú fijo, si aplica.
MainFood	varchar	20	No	Nombre del plato principal incluido en el menú fijo.
SideDish	varchar	20	Sí	Nombre del acompañamiento incluido en el menú fijo, si aplica.
Salad	varchar	20	Sí	Nombre de la ensalada incluida en el menú fijo, si aplica.
Drink	varchar	20	No	Nombre de la bebida incluida en el menú fijo.
StudentPrice	int	N/A	No	Precio del menú fijo para estudiantes.
GeneralPrice	int	N/A	No	Precio general del menú fijo.
TypeMenu	char	15	Sí	Tipo de menú (e.g., Desayuno, Comida).

Tabla 38. Diccionario de datos - SetMenu

Tabla: User

Columna	Tipo de Dato	Longitud	Permite Nulos	Descripción
id	int	N/A	No	Identificador único del usuario.
Email	varchar	255	No	Correo electrónico del usuario.
Password	varchar	61	No	Contraseña del usuario.

Tabla 39. Diccionario de datos - User

Apéndice C Tareas realizadas

Untitled Gantt Project

21 may 2024

<http://>

Encargado del proyecto

Fechas de inicio y fin del proyecto

Paloma Osiris Báez Lara

8 feb 2024 - 14 jun 2024

Progreso

98%

Tarea

206

Recursos

4

Tarea

2

Nombre	Fecha de inicio	Fecha de fin	Progreso	Costo
Comedor Komalli	08/02/24	13/06/24	98	35.810,5
Análisis de requerimientos	08/02/24	29/02/24	80	1.496
Descripción de la problemática	08/02/24	08/02/24	100	374
Entrevista	16/02/24	16/02/24	100	374
Requisitos funcionales y no funcionales	21/02/24	21/02/24	100	374
Identificación de casos de uso	27/02/24	27/02/24	100	280,5
Diccionario de datos	29/02/24	29/02/24	0	93,5
Diseño	29/02/24	04/03/24	100	467,5
Diagrama de casos de uso	29/02/24	29/02/24	100	93,5
Diagrama de despliegue	01/03/24	01/03/24	100	93,5
Diagrama de dominio	01/03/24	01/03/24	100	93,5
Diagrama de clases	01/03/24	01/03/24	100	93,5
Diagrama de paquetes	04/03/24	04/03/24	100	93,5
Construcción	05/03/24	11/03/24	100	1.496
Definición del estandar	05/03/24	05/03/24	100	374
Modelo relacional	05/03/24	05/03/24	100	93,5
Creación de la base de datos	05/03/24	05/03/24	100	93,5
Creación de login	06/03/24	08/03/24	100	561
Prototipo de alta fidelidad	06/03/24	06/03/24	100	93,5
Codificación	07/03/24	07/03/24	100	93,5
Pruebas de unidad	08/03/24	08/03/24	100	93,5
Creación del menú	08/03/24	11/03/24	100	374
Prototipo de alta fidelidad	08/03/24	08/03/24	100	93,5
Codificación	11/03/24	11/03/24	100	93,5
CU-01	04/03/24	11/03/24	100	1.122
Prototipo de alta fidelidad	04/03/24	04/03/24	100	93,5
Descripción de caso de uso	04/03/24	04/03/24	100	93,5
Diagrama de robustez	05/03/24	05/03/24	100	93,5
Diagrama de secuencia	05/03/24	05/03/24	100	93,5
Codificación	06/03/24	06/03/24	100	93,5
Pruebas unitarias	11/03/24	11/03/24	100	93,5
CU-02	04/03/24	11/03/24	100	1.122
Prototipos de alta fidelidad	05/03/24	05/03/24	100	93,5

Tarea

Nombre	Fecha de inicio	Fecha de fin	Progreso	Costo
Descripción de caso de uso	04/03/24	04/03/24	100	93,5
Diagrama de robustez	05/03/24	05/03/24	100	93,5
Diagrama de secuencia	05/03/24	05/03/24	100	93,5
Codificación	06/03/24	06/03/24	100	93,5
Pruebas unitarias	11/03/24	11/03/24	100	93,5
CU-03	04/03/24	11/03/24	100	1.122
Prototipos de alta fidelidad	04/03/24	04/03/24	100	93,5
Descripción de caso de uso	04/03/24	04/03/24	100	93,5
Diagrama de robustez	05/03/24	05/03/24	100	93,5
Diagrama de secuencia	05/03/24	05/03/24	100	93,5
Codificación	06/03/24	06/03/24	100	93,5
Pruebas unitarias	11/03/24	11/03/24	100	93,5
CU-04	04/03/24	11/03/24	100	1.122
Prototipos de alta fidelidad	04/03/24	04/03/24	100	93,5
Descripción de caso de uso	04/03/24	04/03/24	100	93,5
Diagrama de robustez	05/03/24	05/03/24	100	93,5
Diagrama de secuencia	05/03/24	05/03/24	100	93,5
Codificación	06/03/24	06/03/24	100	93,5
Pruebas unitarias	11/03/24	11/03/24	100	93,5
CU-05	13/03/24	19/03/24	100	1.028,5
Prototipo de alta fidelidad	13/03/24	13/03/24	100	93,5
Descripción de caso de uso	13/03/24	13/03/24	100	93,5
Diagrama de robustez	14/03/24	14/03/24	100	93,5
Diagrama de secuencia	14/03/24	14/03/24	100	93,5
Codificación	15/03/24	15/03/24	100	93,5
Pruebas unitarias	19/03/24	19/03/24	100	93,5
CU-06	13/03/24	19/03/24	100	1.028,5
Prototipo de alta fidelidad	13/03/24	13/03/24	100	93,5
Descripción de caso de uso	13/03/24	13/03/24	100	93,5
Diagrama de robustez	14/03/24	14/03/24	100	93,5
Diagrama de secuencia	14/03/24	14/03/24	100	93,5
Codificación	15/03/24	15/03/24	100	93,5
Pruebas unitarias	19/03/24	19/03/24	100	93,5
CU-07	13/03/24	19/03/24	100	1.028,5
Prototipo de alta fidelidad	13/03/24	13/03/24	100	93,5

Tarea

Nombre	Fecha de inicio	Fecha de fin	Progreso	Costo
Descripción de caso de uso	13/03/24	13/03/24	100	93,5
Diagrama de robustez	14/03/24	14/03/24	100	93,5
Diagrama de secuencia	14/03/24	14/03/24	100	93,5
Codificación	15/03/24	15/03/24	100	93,5
Pruebas unitarias	19/03/24	19/03/24	100	93,5
CU-08	13/03/24	19/03/24	100	1.028,5
Prototipo de alta fidelidad	13/03/24	13/03/24	100	93,5
Descripción de caso de uso	13/03/24	13/03/24	100	93,5
Diagrama de robustez	14/03/24	14/03/24	100	93,5
Diagrama de secuencia	14/03/24	14/03/24	100	93,5
Codificación	15/03/24	15/03/24	100	93,5
Pruebas unitarias	19/03/24	19/03/24	100	93,5
CU-09	21/03/24	27/03/24	100	1.028,5
Prototipo de alta fidelidad	21/03/24	21/03/24	100	93,5
Descripción de caso de uso	21/03/24	21/03/24	100	93,5
Diagrama de robustez	22/03/24	22/03/24	100	93,5
Diagrama de secuencia	22/03/24	22/03/24	100	93,5
Codificación	25/03/24	25/03/24	100	93,5
Pruebas unitarias	27/03/24	27/03/24	100	93,5
CU-10	21/03/24	25/03/24	100	748
Prototipo de alta fidelidad	21/03/24	21/03/24	100	93,5
Descripción de caso de uso	21/03/24	21/03/24	100	93,5
Diagrama de robustez	22/03/24	22/03/24	100	93,5
Diagrama de secuencia	22/03/24	22/03/24	100	93,5
Codificación	25/03/24	25/03/24	100	93,5
CU-11	21/03/24	27/03/24	100	1.028,5
Prototipo de alta fidelidad	21/03/24	21/03/24	100	93,5
Descripción de caso de uso	21/03/24	21/03/24	100	93,5
Diagrama de robustez	22/03/24	22/03/24	100	93,5
Diagrama de secuencia	22/03/24	22/03/24	100	93,5
Codificación	25/03/24	25/03/24	100	93,5
Pruebas unitarias	27/03/24	27/03/24	100	93,5
CU-12	29/03/24	05/04/24	100	1.122
Prototipo de alta fidelidad	29/03/24	29/03/24	100	93,5

Tarea

Nombre	Fecha de inicio	Fecha de fin	Progreso	Costo
Descripción de caso de uso	29/03/24	29/03/24	100	93,5
Diagrama de robustez	01/04/24	01/04/24	100	93,5
Diagrama de secuencia	01/04/24	01/04/24	100	93,5
Codificación	02/04/24	02/04/24	100	93,5
Pruebas unitarias	05/04/24	05/04/24	100	93,5
CU-13	29/03/24	05/04/24	100	1.122
Prototipo de alta fidelidad	29/03/24	29/03/24	100	93,5
Descripción de caso de uso	29/03/24	29/03/24	100	93,5
Diagrama de robustez	01/04/24	01/04/24	100	93,5
Diagrama de secuencia	01/04/24	01/04/24	100	93,5
Codificación	02/04/24	02/04/24	100	93,5
Pruebas unitarias	05/04/24	05/04/24	100	93,5
CU-14	29/03/24	05/04/24	100	1.122
Prototipo de alta fidelidad	29/03/24	29/03/24	100	93,5
Descripción de caso de uso	29/03/24	29/03/24	100	93,5
Diagrama de robustez	01/04/24	01/04/24	100	93,5
Diagrama de secuencia	01/04/24	01/04/24	100	93,5
Codificación	02/04/24	02/04/24	100	93,5
Pruebas unitarias	05/04/24	05/04/24	100	93,5
CU-15	29/03/24	05/04/24	100	1.122
Prototipo de alta fidelidad	29/03/24	29/03/24	100	93,5
Descripción de caso de uso	29/03/24	29/03/24	100	93,5
Diagrama de robustez	01/04/24	01/04/24	100	93,5
Diagrama de secuencia	01/04/24	01/04/24	100	93,5
Codificación	02/04/24	02/04/24	100	93,5
Pruebas unitarias	05/04/24	05/04/24	100	93,5
CU-16	09/04/24	15/04/24	100	935
Prototipo de alta fidelidad	09/04/24	09/04/24	100	93,5
Descripción de caso de uso	09/04/24	09/04/24	100	0
Diagrama de robustez	10/04/24	10/04/24	100	93,5
Diagrama de secuencia	10/04/24	10/04/24	100	93,5
Codificación	11/04/24	11/04/24	100	93,5
Pruebas unitarias	15/04/24	15/04/24	100	93,5
CU-17	09/04/24	15/04/24	100	1.028,5
Prototipo de alta fidelidad	09/04/24	09/04/24	100	93,5

Tarea

Nombre	Fecha de inicio	Fecha de fin	Progreso	Costo
Descripción de caso de uso	09/04/24	09/04/24	100	93,5
Diagrama de robustez	10/04/24	10/04/24	100	93,5
Diagrama de secuencia	10/04/24	10/04/24	100	93,5
Codificación	11/04/24	11/04/24	100	93,5
Pruebas unitarias	15/04/24	15/04/24	100	93,5
CU-18	09/04/24	15/04/24	100	1.028,5
Prototipo de alta fidelidad	09/04/24	09/04/24	100	93,5
Descripción de caso de uso	09/04/24	09/04/24	100	93,5
Diagrama de robustez	10/04/24	10/04/24	100	93,5
Diagrama de secuencia	10/04/24	10/04/24	100	93,5
Codificación	11/04/24	11/04/24	100	93,5
Pruebas unitarias	15/04/24	15/04/24	100	93,5
CU-19	09/04/24	11/04/24	100	748
Prototipo de alta fidelidad	09/04/24	09/04/24	100	93,5
Descripción de caso de uso	09/04/24	09/04/24	100	93,5
Diagrama de robustez	10/04/24	10/04/24	100	93,5
Diagrama de secuencia	10/04/24	10/04/24	100	93,5
Codificación	11/04/24	11/04/24	100	93,5
CU-20	17/04/24	24/04/24	100	1.122
Prototipo de alta fidelidad	17/04/24	17/04/24	100	93,5
Descripción de caso de uso	17/04/24	17/04/24	100	93,5
Diagrama de robustez	18/04/24	18/04/24	100	93,5
Diagrama de secuencia	18/04/24	18/04/24	100	93,5
Codificación	19/04/24	19/04/24	100	93,5
Pruebas unitarias	24/04/24	24/04/24	100	93,5
CU-21	17/04/24	24/04/24	100	1.122
Prototipo de alta fidelidad	17/04/24	17/04/24	100	93,5
Descripción de caso de uso	17/04/24	17/04/24	100	93,5
Diagrama de robustez	18/04/24	18/04/24	100	93,5
Diagrama de secuencia	18/04/24	18/04/24	100	93,5
Codificación	19/04/24	19/04/24	100	93,5
Pruebas unitarias	24/04/24	24/04/24	100	93,5
CU-22	17/04/24	24/04/24	100	1.122
Prototipo de alta fidelidad	17/04/24	17/04/24	100	93,5

Tarea

Nombre	Fecha de inicio	Fecha de fin	Progreso	Costo
Descripción de caso de uso	17/04/24	17/04/24	100	93,5
Diagrama de robustez	18/04/24	18/04/24	100	93,5
Diagrama de secuencia	18/04/24	18/04/24	100	93,5
Codificación	19/04/24	19/04/24	100	93,5
Pruebas unitarias	24/04/24	24/04/24	100	93,5
CU-23	17/04/24	19/04/24	100	748
Prototipo de alta fidelidad	17/04/24	17/04/24	100	93,5
Descripción de caso de uso	17/04/24	17/04/24	100	93,5
Diagrama de robustez	18/04/24	18/04/24	100	93,5
Diagrama de secuencia	18/04/24	18/04/24	100	93,5
Codificación	19/04/24	19/04/24	100	93,5
CU-24	26/04/24	30/04/24	100	841,5
Prototipo de alta fidelidad	26/04/24	26/04/24	100	93,5
Descripción de caso de uso	26/04/24	26/04/24	100	93,5
Diagrama de robustez	29/04/24	29/04/24	100	93,5
Diagrama de secuencia	29/04/24	29/04/24	100	93,5
Codificación	29/04/24	29/04/24	100	93,5
Pruebas unitarias	30/04/24	30/04/24	100	93,5
CU-25	03/05/24	06/05/24	100	2.805
Prototipo de alta fidelidad	01/05/24	01/05/24	100	93,5
Descripción de caso de uso	01/05/24	01/05/24	100	93,5
Diagrama de robustez	02/05/24	02/05/24	100	93,5
Diagrama de secuencia	02/05/24	02/05/24	100	93,5
Codificación	03/05/24	03/05/24	100	93,5
Pruebas unitarias	06/05/24	06/05/24	100	93,5
CU-26	08/05/24	09/05/24	100	2.711,5
Prototipo de alta fidelidad	07/05/24	07/05/24	100	93,5
Descripción de caso de uso	07/05/24	07/05/24	100	93,5
Diagrama de robustez	08/05/24	08/05/24	100	93,5
Diagrama de secuencia	08/05/24	08/05/24	100	93,5
Codificación	09/05/24	09/05/24	100	93,5
Pruebas de software	03/06/24	04/06/24	50	1.496
Pruebas de integración	03/06/24	03/06/24	50	374
Pruebas de sistema	04/06/24	04/06/24	50	374
Documentación del sistema	12/06/24	13/06/24	90	748

Recursos

8

Nombre	Función
Alejandro Sánchez Marín	Desarrollador
Ares Judda Rivera Soto	Desarrollador
Paloma Osiris Báez Lara	Encargado del proyecto
Miriam Ramírez Zárate	Desarrollador

Diagrama de Gantt

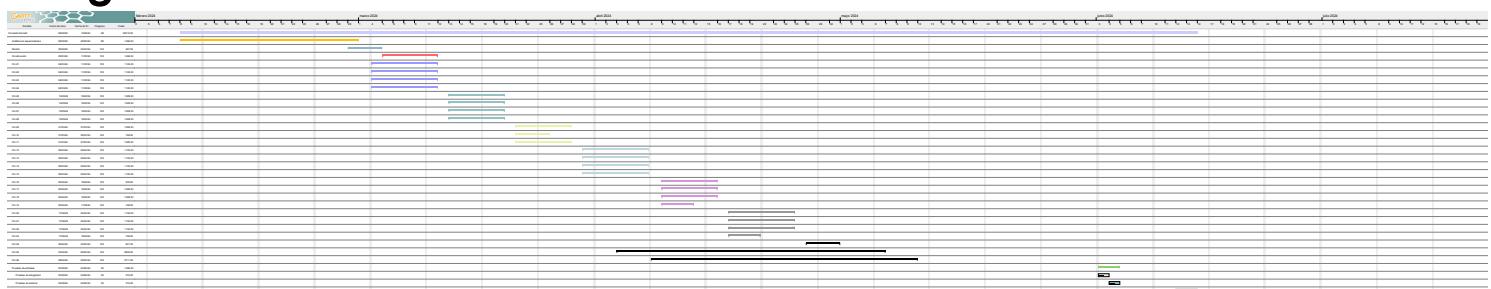
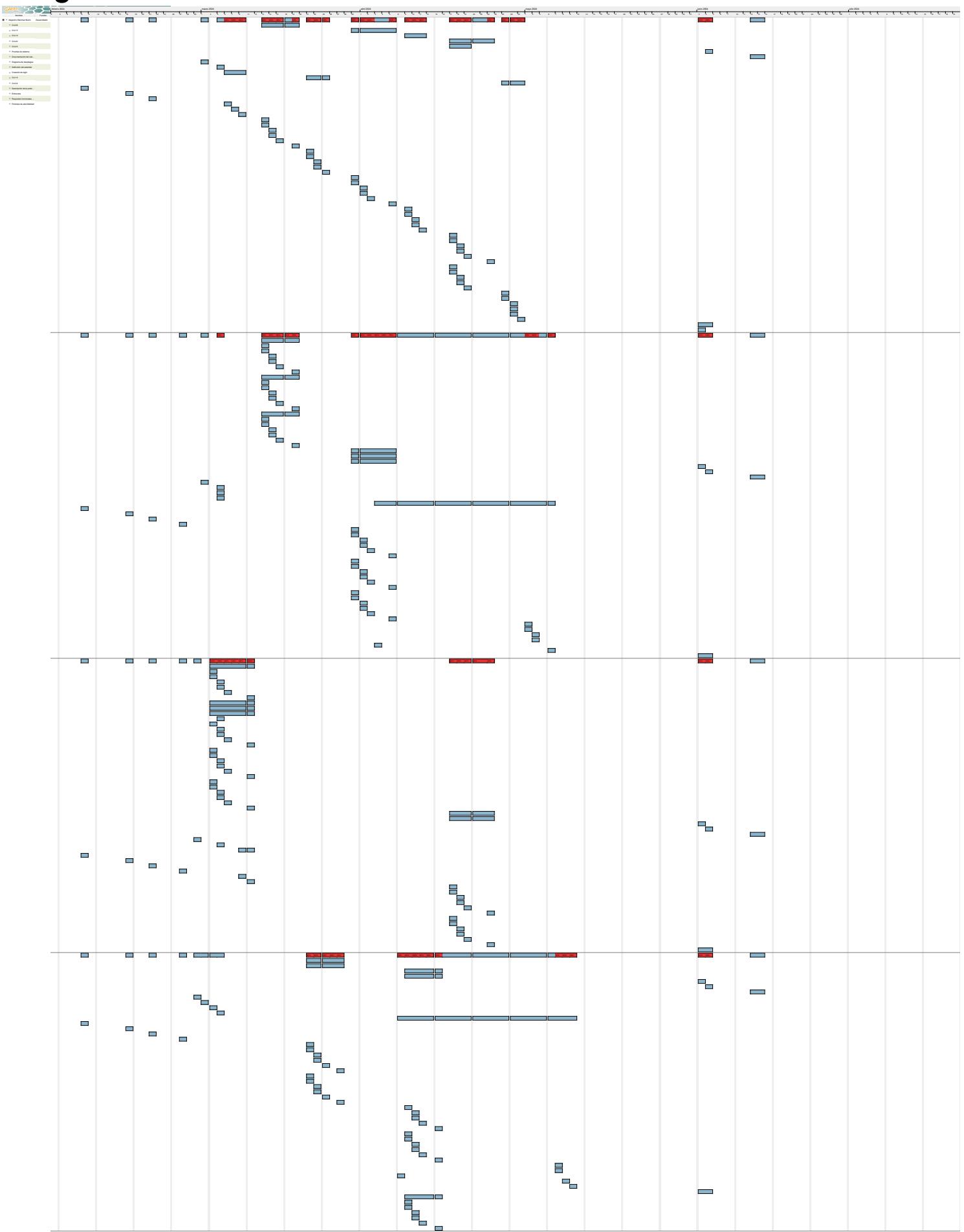


Diagrama de recursos



Apéndice D. Estándar de codificación

Desarrollo de software

Estándar de codificación

Equipo 4

Miriam Ramírez Zárate

Ares Judda Rivera Soto

Alejandro Sánchez Marín

Paloma Osiris Báez Lara

29/03/2024

Contenido

Introducción.....	2
Propósito	2
Reglas o convención de nombres.....	3
Métodos	3
Variables	3
Clases.....	4
User controls	4
Constantes	5
Nombres de componentes.....	5
Metodos de pruebas.....	6
Controladores.....	7
Organización de archivos	8
Definición de métodos, clases y variables	8
Organización de código.....	9
Idioma	9
Llaves.....	9
Comentarios	9
Manejo de excepciones.....	10
Prácticas de seguridad	11
Referencias	12

Introducción

Las prácticas recomendadas en programación abarcan un conjunto de métodos, conceptos y normas empleados para crear código de excelente calidad y que pueda ser mantenido con facilidad. Estas pautas abordan aspectos clave como la legibilidad, el rendimiento, la capacidad de crecimiento, la seguridad y la facilidad de mantenimiento del código.

La importancia de aplicar prácticas óptimas en programación es diversa:

1. Simplifican la comprensión del código: Al utilizar convenciones y estructuras comunes y bien definidas, se vuelve más fácil para otros programadores entender el código escrito.
2. Mejoran la calidad del código: Las prácticas de programación adecuadas ayudan a prevenir errores y problemas, lo que resulta en un software más sólido y estable.
3. Incrementan la eficiencia: Al seguir prácticas como la optimización del código y la reutilización de componentes, se logra aumentar la velocidad de ejecución y reducir los recursos necesarios.
4. Promueven la escalabilidad: Al emplear patrones de diseño y arquitecturas bien establecidas, es posible crear software escalable, capaz de expandirse y evolucionar con el tiempo.
5. Garantizan la seguridad: Las prácticas adecuadas en programación contribuyen a prevenir vulnerabilidades y ataques maliciosos. En resumen, las buenas prácticas en programación son esenciales para producir software de alta calidad, que sea sencillo de mantener y que tenga la capacidad de crecer. Esto se traduce en un software más eficiente y seguro.
6. Detección temprana de fallas: Al buscar cumplir con los estándares es más sencillo detectar posibles errores desde la revisión de código, evitando que esos problemas lleguen a producción.

Propósito

El propósito principal es instaurar la norma que se seguirá a lo largo de todo el proceso de desarrollo del proyecto relacionado con los fundamentos de construcción de software en el lenguaje C#. El objetivo es dirigirse hacia la creación de un programa que combine tanto la excelencia como la sofisticación.

Para lograr este fin, se aplicarán las siguientes pautas:

1. Uniformidad en la estructura del código: Se mantendrá una coherencia en la estructura del código, empleando una indentación clara y constante. Además, se respetarán las convenciones de nomenclatura apropiadas, utilizando nombres descriptivos para variables, métodos y clases.
2. Modularidad y cohesión: El código se organizará en módulos lógicos y cohesionados, cada uno con una responsabilidad única y bien definida. Esto facilitará la comprensión y el mantenimiento del programa.
3. Incorporación de comentarios: Se incluirán comentarios claros y concisos en el código para explicar la funcionalidad y la lógica detrás de las implementaciones.
4. Prevención de duplicación de código: Se fomentará la reutilización de código mediante enfoques como la creación de funciones y clases genéricas, así como la

utilización de patrones de diseño. Esto reducirá la redundancia y mejorará la capacidad de mantenimiento del código.

5. Pruebas unitarias: Se implementarán pruebas unitarias para confirmar el correcto funcionamiento de cada componente y detectar posibles errores en etapas tempranas. Esto garantizará la calidad y la robustez del programa.
6. Manejo adecuado de excepciones: Se aplicará un manejo cuidadoso de las excepciones para asegurar que el programa sea resistente y no se vea interrumpido por fallos no controlados.

En resumen, siguiendo estas normas y prácticas de desarrollo, nuestro propósito es crear un sistema de gestión para la cafetería Komalli en C# que refleje distinción y sofisticación. Esto se traducirá en un código bien estructurado, de fácil comprensión, mantenible y eficiente.

Reglas o convención de nombres

Métodos

Los métodos tendrán nombres descriptivos usando verbos en infinitivo de acuerdo con lo que se realizará, así como se utilizará el estilo de escritura de PascalCase. No se permiten caracteres especiales.

Uso correcto:

```
public string CreateRoom(string gamertag) {  
}
```

Uso no correcto:

```
public string generateId() {  
}
```

Variables

Las variables se nombrarán usando "snake_case_with_underscore_prefix" si se trata de variables privadas, las públicas se hará llamándolas con la primera letra en mayúscula y para esto se hará usando sustantivos y palabras descriptivas de acuerdo con el uso que tendrán. Las excepciones son los ciclos for los cuales está permitido el uso de i, j, k, l siempre y cuando se siga ese orden si es que se anida más de un ciclo.

Uso correcto:

```
public int IdUser { get; set;  
  
private string _gamertag;  
  
for (int i = 0; i < 10; i++) {  
    for (int j = 0; j < 10; j++) {  
        for (int k = 0; k < 10; k++) {  
            for (int l = 0; l < 10; l++) {
```

```
        }
    }
}
```

Uso no correcto:

```
private string Report;
public string _report;

for (int j = 0; j < 10; j++) {
    for (int i = 0; i < 10; i++) {

    }
}
```

Clases

Las clases estarán definidas usando el CamelCase y tendrán nombres descriptivos usando sustantivos. Debe evitarse el uso de acrónimos o abreviaturas, salvo en aquellos casos en los que dicha abreviatura sea más utilizada que la palabra que representa (URL, HTTP, etc.).

Uso correcto:

```
public class Friend {
```

Uso no correcto:

```
public class getStudents {
```

User controls

Los user controls estarán definidas usando el CamelCase y tendrán nombres descriptivos usando sustantivos. Debe evitarse el uso de acrónimos o abreviaturas, salvo en aquellos casos en los que dicha abreviatura sea más utilizada que la palabra que representa (URL, HTTP, etc.).

Uso correcto:

```
Public partial class FriendList : UserControl{
```

Uso no correcto:

```
public partial class playersgame : UserControl{
```

Constantes

Todas las constantes se escribirán en mayúsculas, de ser frases nominales se separarán las palabras entre si con guiones bajos (_).

Uso correcto:

```
const string CHARACTERS = "0123456789";
```

Uso no correcto:

```
const string image_profile = "/Komalli;component/Resources/Images/Profile.png";
```

Nombres de componentes

Los componentes se nombrarán comenzando en minúscula y todos tendrán el prefijo del componente que se trate y seguido será algo descriptivo de lo que realice.

Uso correcto:

```
x:Name="txbChat"
```

Uso no correcto:

```
x:Name="textBoxChat"
```

Control	Prefijo
Label	lbl
TextBox	txt
DataGridView	dtg
Button	btn
ImageButton	imb
Hyperlink	hlk
DropDownList	ddl
ListBox	lst
DataList	dtl
Repeater	rep
Checkbox	chk
CheckboxList	cbl
RadioButton	rbt
RadioButtonlist	rbl
Image	img
Panel	pan
PlaceHolder	phd
Table	tbl
Validators	val

Stack panel – stp

Text Block – txb

Metodos de pruebas

Los métodos de prueba seguirán la convención CamelCase y se denominarán según el nombre del método que se está probando, seguido por un guion bajo y el tipo de prueba, ya sea "Successful," "Failed" o "Exception." En caso de que se realicen múltiples pruebas del mismo tipo en un método específico, después del tipo de prueba se especificará, mediante CamelCase, el número correspondiente (por ejemplo, "SuccessfulOne"). En situaciones en las que solo se realice una prueba de un tipo específico, no será necesario indicar el número de prueba.

Uso Correcto:

```
public void AddUserAccount_SuccessfullOne () {
```

```
}
```

```
public void AddUserAccount_SucessfullTwo() {
```

```
}
```

```
public void UpdateInfoUser_Failed( ){
```

```
}
```

```
public void ValidateUser_Exception () {
```

```
}
```

Uso no correcto:

```
public void AddUserAccount_Sucessfull2() {
```

```
}
```

```
public void UpdateInfoUser-failed( ){
```

```
}
```

Controladores

Los controladores seguirán la convención de CamelCase y se denominaran según el tipo de evento que desencadenan, seguido de un nombre descriptivo sobre la función que realizan.

Uso correcto:

```
private void ClickPlay (object sender, RoutedEventArgs e){
```

```
}
```

Uso no correcto:

```
private void Options (object sender, RoutedEventArgs e){
```

```
}
```

Organización de archivos

Se usará el patrón de diseño MVC, por ello se definirá 3 capas llamadas Model, View y Controller.

Dentro de la capa de Model se especificará las clases que hagan referencia a la estructura de los datos y la lógica del negocio. En la carpeta de View estará todo aquel archivo XAML que contenga el diseño de las interfaces de usuario. Y, por último, en la capa de Controller contendrá la lógica y manejará las interacciones del usuario.

Los nombres de los archivos y clases que se definan en cada capa serán simples, es decir, serán sustantivos y el nombre será descriptivo según lo que se realice.



Definición de métodos, clases y variables

Las clases y métodos serán de acceso público, sin embargo, las variables serán definidas como privadas, usando la estructura de nombrado "snake_case_with_underscore_prefix", para el uso de sus getters y setters los cuales serán en forma de propiedades, serán públicos y seguirán la estructura de nombrado CamelCase.

Uso correcto:

```
private string _gamertag;
public string Gamertag {
    get { return _gamertag; }
    set {
        if (_gamertag != value) {
            _gamertag = value;
            OnPropertyChanged(nameof(Gamertag));
        }
    }
}
```

Uso no correcto:

```
private string gamertag;
public string _Gamertag {
    get { return gamertag; }
    set {
        if (gamertag != value) {
            gamertag = value;
            OnPropertyChanged(nameof(_Gamertag));
        }
    }
}
```

Organización de código

Idioma

El idioma que se utilizará para el desarrollo del código será el inglés.

Llaves

El estilo por emplear para las llaves es el llamado OTBS, es decir, al declarar una clase, método, etc. Las funciones tienen llaves de apertura en la misma línea separadas por un espacio y las llaves que cierran si estarán en su propia línea. Las propiedades que son simples se le puede declarar las llaves de apertura y final sobre la misma línea.

Uso correcto:

```
private void LoginPasswordBox(object sender, KeyEventArgs e) {  
    if (e.Key == Key.Enter) {  
        SignIn(sender, e);  
    }  
}
```

Uso no correcto:

```
private void LoginPasswordBox(object sender, KeyEventArgs e)  
{  
    if (e.Key == Key.Enter)  
    {  
        SignIn(sender, e);  
    }  
}
```

Comentarios

Se usarán comentarios de una sola línea (//) cuando se trate de explicaciones breves. Cuando sean extensas se usarán los comentarios /* */. Se colocará un espacio entre el delimitador del

comentario y el texto del comentario. Únicamente se usarán los comentarios cuando sea necesarios explicar el fundamento del código, puesto que el código se debe de explicar solo. Uso correcto:

```
// Hola mundo  
  
/*      ngerbread I love t ram su jujubes sweet roll. T ram su marz  
 *      bear claw loll pop pastry dan sh. Marshmallow sesame snaps  
 *      cheesecake cotton candy. I love macaroon muff n chocolate cake  
 */
```

Uso no correcto:

```
//Hola mundo  
/*Callback */
```

El compilador de C# combina la estructura del código C# con el texto de los comentarios en un único documento XML. La estructura de los comentarios ocupados para la generación de

documentación de API estará definida de la siguiente manera:

Uso correcto:

```
/***
 * <summary>
 * Genera un identificador a las salas que se crean.
 * </summary>
 * <returns>Retorna una cadena que es el identificador de la sala</returns>
 */

/// <summary>
/// Interfaz que contiene los metodos que se encarga de la funcionalidad de la
/// sala
/// </summary>
```

Uso no correcto:

```
/// <summary> Clase de ejemplo</summary>
///

/*<summary>
 *           * This method is used to generate a unique identifier for the guest
player.
 *           * </summary>
 *           * <returns>Unique identifier for the guest
player.</returns>
 *           */

```

Manejo de excepciones

La instrucción "try" debe ser seguida por una llave de apertura en su propia línea. A continuación, el cuerpo de la sentencia deberá ser colocado, seguido de un salto de línea antes de la llave de cierre correspondiente. Posteriormente, se realizará la captura de la excepción, asegurándose de que la sentencia "catch" esté en una línea separada. Es imperativo incluir un bloque "finally" para la destrucción de objetos no necesarios. Es esencial evitar la creación de bloques "try-catch" vacíos.

Se debe evitar descartar excepciones, incluso si se considera improbable que ocurra algún fallo en el sistema. Si es necesario dejar una excepción sin capturar, se debe proporcionar una justificación en un comentario que explique el motivo de esta decisión. Se recomienda evitar el uso de excepciones genéricas (como "Exception") debido a que esto capturaría todas las excepciones, dificultando la identificación precisa del error en situaciones donde solo se requiere manejar una excepción específica.

Ejemplo:

```
public string CreateRoom(string gamertag) {
```

```

try {
    string idRoom;
    do {
        idRoom = GenerateId();
    } while (_rooms.Contains(idRoom));
    _rooms.Add(idRoom);
    return idRoom;
} catch (CommunicationException ex) {
    MessageBox.Show($"Error al crear la sala: {ex.Message}");
    return null;
}
}

```

Prácticas de seguridad

Consultas parametrizadas: El uso de consultas parametrizadas ayuda a prevenir la inyección de SQL, que es un vector común de ataque. Al usar consultas parametrizadas, los valores proporcionados por los usuarios se tratan como datos y no como parte de la consulta SQL en sí.

Uso correcto:

```

string query = "INSERT INTO Usuarios (Nombre, Apellido) VALUES (@Nombre,
@Apellido)";
using (SqlConnection connection = new SqlConnection(connectionString)) {
    using (SqlCommand command = new SqlCommand(query, connection)) {
        command.Parameters.AddWithValue("@Nombre", nombreUsuario);
        command.Parameters.AddWithValue("@Apellido", apellidoUsuario);
        connection.Open(); command.ExecuteNonQuery();
    }
}

```

Uso no correcto:

```

string query = "INSERT INTO Usuarios (Nombre, Apellido) VALUES
('" + nombreUsuario + "', '" + apellidoUsuario + "')";
using (SqlConnection connection = new SqlConnection(connectionString))
{ using (SqlCommand command = new SqlCommand(query, connection)) {
    connection.Open(); command.ExecuteNonQuery();
}
}

```

Encriptación: Se utilizará el cifrado adecuado para proteger datos sensibles, ya sea en reposo (almacenamiento) o en tránsito (comunicaciones). Utilizando algoritmos de encriptación fuertes, la realización de hash seguras como lo es bcrypt.

Ejemplo:

```

public int AddUserAccount(UserModel userModel) {
    try {
        using (var context = new ChineseCheckersEntities())
        using (var transaction = context.Database.BeginTransaction()) {
            var userAccountGame = new UserAccount {
                GamertagUser = userModel.Gamertag,
                Email = userModel.Email,
                Password = BCrypt.Net.BCrypt.HashPassword(userModel.Password),
                ImageProfile = userModel.ImageProfile,
            };
        }
    }
}

```

```
        context.UserAccount.Add(userAccountGame);
    try {
        context.SaveChanges();
    } catch (DbUpdateException ex) {
        transaction.Rollback();
        MessageBox.Show($"Error de concurrencia al agregar el usuario: {ex.Message}");
        return -1;
    }
    transaction.Commit();
}
} catch (EntityException ex) {
    MessageBox.Show($"Error al agregar la cuenta de usuario: {ex.Message}");
    return -1;
}
return 1;
}
```

Referencias

- Meneses, S. (2021, October 5). La Importancia de los Estándares de Código. The Dojo MX Blog. <https://blog.thedojo.mx/2021/10/05/estandares-de-calidad-en-el-software.html>
- Indentación estilo K&R y variantes. (n.d.). Com.ar. Retrieved August 24, 2023, from http://www.luchonet.com.ar/aed/?page_id=165
- BillWagner. (n.d.). Convenciones de codificación de C# de documentación de .NET. Microsoft.com.
- Retrieved August 24, 2023, from <https://learn.microsoft.com/es-es/dotnet/csharp/fundamentals/coding-style/coding-conventions>
- BillWagner. (n.d.-b). Instrucciones de control de excepciones: throw y try, catch, finally. Microsoft.com. Retrieved August 24, 2023, from <https://learn.microsoft.com/es-es/dotnet/csharp/language-reference/statements/exception-handling-statements>