



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

OPTIMIZACIÓN DE PREPARACIÓN DE PLATILLOS PARA RESTAURANTES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

JING HENG LUO ZENG

PROFESOR GUÍA:
FRANCISCO GUTIÉRREZ FIGUEROA

MIEMBROS DE LA COMISIÓN:
NELSON BALOIAN TATARYAN
HUGO ANDRÉS MORA RIQUELME

SANTIAGO DE CHILE
2023

Resumen

En la actualidad, existen aplicaciones de gestión de restaurantes conocidas como Terminales de Punto de Venta (TPV). Algunas de estas aplicaciones permiten ingresar pedidos a través de dispositivos manejados por el personal y mostrarlos en la cocina, mientras que otras soluciones más completas incluso instalan dispositivos de gran tamaño para que los clientes interactúen directamente con ellos y se muestren los pedidos en la cocina. Sin embargo, ninguna solución existente que optimiza el orden de preparación de los artículos según su prioridad para cada estación de trabajo. Por esta razón, el objetivo de esta memoria fue desarrollar un sistema TPV que permita el ingreso sencillo de pedidos y muestre los artículos en el orden óptimo de preparación, optimizando así el tiempo de cocción de alimentos.

Mediante entrevistas con expertos de dominio, se definió un conjunto de conceptos y reglas que condujeron a la creación de un modelo de dominio. Este modelo permitió construir un prototipo de una aplicación web TPV que permite al usuario ingresar los pedidos, que luego se muestran de manera clara en la cocina. El prototipo final cumple con los requerimientos expresados por expertos de dominio con más de 10 años de experiencia en el rubro, a pesar de que las interfaces de usuario no sean muy intuitivas, poseen una baja curva de aprendizaje.

El desarrollo del prototipo fue evaluado en tres fases. En la primera, se contó con usuarios con experiencia en el rubro para verificar la usabilidad de la vista de toma de pedidos del sistema y estudiar la usabilidad del prototipo realizado. La prueba consistió en realizar tareas dentro de la vista, midiendo ciertas métricas, como el número de usuarios que cometieron el mismo error. De esta fase, se identificaron oportunidades de mejora para algunas funcionalidades de la vista de toma de pedidos.

La segunda fase de la evaluación se llevó a cabo en un ambiente de laboratorio para verificar si el funcionamiento de las colas era el esperado y así comprobar que el sistema al menos pudiese manejar algunos casos comunes dentro del rubro. Esta prueba consistió en realizar una serie de acciones y comparar los resultados con los resultados esperados.

Finalmente, en la última fase de la evaluación, se realizaron pruebas en un ambiente real, donde se instaló el sistema en un restaurante de comida china para estudiar y comparar los resultados entre el caso con el sistema desarrollado y el caso del restaurante sin intervención. Para ello, se utilizó diariamente la aplicación en el restaurante, enfrentando situaciones reales y midiendo distintas métricas. Luego de unos días de pruebas, se midieron las mismas métricas en el restaurante sin intervención, para luego comparar los resultados y obtener conclusiones. A partir de esta fase, se identificaron oportunidades para mejorar las interfaces y posibles funcionalidades que aumenten el valor del sistema.

Agradecimientos

Primero que todo, quiero agradecer a mi familia que me han apoyado durante toda mi formación. A mis padres por estar siempre ahí cuando necesitaba algo, por sacrificarse tanto para poder sacarme adelante. También a mi hermana y primos por apoyarme durante el desarrollo de este trabajo, dándome ideas al principio. En particular a mi primo Zehui que fue con quien más hable acerca del desarrollo, siendo de mucha ayuda. También agradezco a mis amigos que me apañaron durante las noches de juego y conversaciones que tuvimos durante toda la carrera, gracias a ellos se me hizo muy intensa y no fue tan estresante.

También quiero agradecer a quienes han hecho posible esta memoria y han dado de su tiempo para poder lograr el mejor trabajo posible. Al profesor Francisco y a mis compañeros Támara y Lung, muchas gracias por aportarme todo su conocimiento y retroalimentación en este proceso. A todos los expertos del dominio que participaron en este proceso, muchas gracias por ayudarme a sentar las bases del proyecto. Y por último agradezco a los usuarios que me ayudaron a probar el sistema y al personal del restaurante, que, sin el aporte de ellos, este trabajo no hubiera tenido el mismo valor.

Tabla de contenido

1. Introducción	1
1.1. Objetivos	2
1.2. Solución Desarrollada	3
1.3. Estructura del documento	4
2. Trabajo relacionado	7
2.1. Aplicaciones existentes	7
2.2. Estado de la práctica	8
3. Desarrollo de la solución	9
3.1. Análisis y Modelado	9
3.1.1. Variables del Modelo de Optimización	9
3.1.2. Prototipo inicial de Modelo de solución	11
3.1.3. Exploración de tecnologías	13
3.1.4. Modelo de datos inicial	14
3.2. Configuración del proyecto inicial	16
3.3. Diseño e implementación de sistema de toma de pedidos	16
3.4. Diseño e implementación de visualización de pedidos	19
3.5. Diseño e implementación de vistas para las colas	23
3.6. Modelo de datos del prototipo final	30
3.7. Estructura general de la aplicación	32
4. Evaluación de la solución	34
4.1. Definición de la evaluación	34
4.2. Protocolo de experimentación	34
4.2.1. Protocolo para vista TPV	34
4.2.2. Protocolo para vistas de información de pedidos	39
4.3. Materiales	49
4.4. Reproducibilidad del estudio	51
5. Resultados	53
5.1. Resultados del experimento de la vista TPV	53
5.1.1. Wilson y Wixon [4]	53
5.1.2. Encuesta Post-Experiencia	55
5.2. Resultados del experimento de vistas de pedidos en ambiente controlado	56
5.2.1. Pruebas controladas	56

5.3.	Resultados del experimento de vistas de pedidos en ambiente real	56
5.3.1.	Datos recopilados en el restaurante con el sistema	57
5.3.2.	Datos recopilados en el restaurante sin el sistema	59
6.	Análisis y discusión de resultados	61
6.1.	Análisis de resultados del experimento 1	61
6.2.	Análisis de resultados del experimento 2	63
6.3.	Análisis de resultados del experimento 3	64
6.4.	Posibles extensiones del sistema	66
7.	Conclusión y Trabajo Futuro	68
	Bibliografía	72
	Anexo A. Instrumentos	74
A.1.	Experimento 1	74
A.2.	Experimento 2	76
A.3.	Experimento 3	84
	Anexo B. Resultados	86
B.1.	Experimento 1	86
B.2.	Experimento 2	87
B.3.	Experimento 3	97

Índice de ilustraciones

3.1. Página por defecto de <i>Django</i> desde dispositivo ejecutor.	14
3.2. Página por defecto de <i>Django</i> desde dispositivo secundario.	14
3.3. Modelo de datos inicial.	15
3.4. Vista de ingreso de artículos (Itactil).	16
3.5. Vista de total de pedido (Itactil).	16
3.6. Mock-up de sistema TPV propio realizado en Figma.	17
3.7. Vista prototipo final sistema TPV de la aplicación.	17
3.8. Vista prototipo final de notas de un artículo en el sistema TPV de la aplicación.	18
3.9. Mock-up 1 de <i>display</i> de pedidos realizado en Figma.	20
3.10. Mock-up 2 de <i>display</i> de pedidos realizado en Figma.	20
3.11. Mock-up 2 de <i>display</i> de pedidos tras presionar un botón de “Listo” realizado en Figma.	21
3.12. Vista prototipo final de pedidos pendientes de la aplicación.	21
3.13. Vista prototipo final de pedidos pendientes (modal cantidad de artículos tras presionar botón “Listo”) de la aplicación.	22
3.14. Vista prototipo final de pedidos pendientes (con un artículo listo) de la aplicación.	22
3.15. Vista prototipo final de pedidos pendientes (con un artículo parcialmente listos) de la aplicación.	22
3.16. Mock-up de <i>display</i> de colas realizado en Figma.	24
3.17. Mock-up de <i>display</i> de colas plegables realizado en Figma.	24
3.18. Vista prototipo final de colas sin seleccionar ninguna estación de la aplicación.	25
3.19. Vista prototipo final de colas seleccionando 4 estaciones de la aplicación.	25
3.20. Vista prototipo final de colas seleccionando 6 estaciones de la aplicación.	26
3.21. Vista prototipo final de colas desde Galaxy Tab A 8” 2019 SM-T290 (1280 x 800 píxeles) de la aplicación.	26
3.22. Funciones del tipo $f(x) = A * x^b$ vs. identidad graficada en <i>GeoGebra</i>	27
3.23. Modelo de datos Final.	31
4.1. ASUS TUF Dash F15 (2022). Fuente: https://www.asus.com	50
4.2. Samsung Galaxy Tab A SM-T290 (2019). Fuente: https://www.samsung.com	51
4.3. iPad (6th generation) (2018). Fuente: https://support.apple.com	51
6.1. Vista sistema actual.	62
6.2. Vista TPV con cambios propuestos de reemplazo de scroll por flechas, cambio de icono de notas y barra de búsqueda de artículos.	62
6.3. Vista propuesta de configuración de fuentes de las vistas de la aplicación.	62

6.4. Vista sistema actual.	63
6.5. Vista TPV con cambio del flujo de modificación de combinaciones.	63
6.6. Vista propuesta de TPV.	65
6.7. Vistas TPV propuesta creación artículos tras presionar el botón de añadir artículo.	65
6.8. Vistas de pedidos propuesta para el manejo de sobras.	66
A.1. Formulario durante intervención experimento 1.	74
A.2. Encuesta post-intervención experimento 1.	75
A.3. Tabla que resume los resultados de las pruebas del experimento 2.	76
A.4. Configuración experimento 2.	76
A.5. Prueba 1 experimento 2.	77
A.6. Prueba 2 experimento 2.	77
A.7. Prueba 3 experimento 2.	78
A.8. Prueba 4 experimento 2.	79
A.9. Prueba 5 experimento 2.	80
A.10. Prueba 6 experimento 2.	81
A.11. Prueba 7 experimento 2.	81
A.12. Prueba 8 experimento 2.	82
A.13. Prueba 9 experimento 2.	83
A.14. Plantilla de la tabla de registro de errores para el caso con el sistema instalado del experimento 3.	84
A.15. Plantilla de la tabla de registro de errores para el caso sin el sistema instalado del experimento 3.	84
A.16. Plantilla de las tablas de recopilación de datos de pedidos experimento 3.	85
B.1. Gráfico de barras de número de errores promedio del experimento 1.	86
B.2. Diagrama de caja de número de errores promedio del experimento 1.	86
B.3. Gráfico de barras de número de ayudas promedio del experimento 1.	87
B.4. Diagrama de caja de número de ayudas promedio del experimento 1.	87
B.7. Tabla de resultados de las pruebas del experimento 2.	87
B.5. Resultados en crudo de los datos recopilados durante la experiencia del experimento 1.	88
B.6. Resultados en crudo de los datos recopilados después de la experiencia del experimento 1.	89
B.8. Resultados de la prueba 1 experimento 2.	90
B.9. Resultados de la prueba 2 experimento 2.	90
B.10. Resultados de la prueba 3 experimento 2.	91
B.11. Resultados de la prueba 4 experimento 2.	92
B.12. Resultados de la prueba 5 experimento 2.	93
B.13. Resultados de la prueba 6 experimento 2.	94
B.14. Resultados de la prueba 7 experimento 2.	94
B.15. Resultados de la prueba 8 experimento 2.	95
B.16. Resultados de la prueba 9 experimento 2.	96
B.17. Gráfico de barras de número de pedidos durante la prueba con sistema del experimento 3.	97

B.18. Diagrama de caja de número de pedidos durante la prueba con sistema del experimento 3.	97
B.19. Gráfico de barras de media de tiempo en segundo que demora un pedido durante la prueba con sistema del experimento 3.	98
B.20. Diagrama de caja de tiempo en segundo que demora un pedido durante la prueba con sistema del experimento 3.	98
B.21. Gráfico de barras de total de artículos por día durante la prueba con sistema del experimento 3.	99
B.22. Gráfico de barras de media de artículos de un pedido por día durante la prueba con sistema del experimento 3.	99
B.23. Diagrama de caja de número de artículos durante la prueba con sistema del experimento 3.	100
B.24. Gráfico de barras de media de artículos distintos de un pedido durante la prueba con sistema del experimento 3.	100
B.25. Diagrama de caja de media de artículos distintos de un pedido durante la prueba con sistema del experimento 3.	101
B.26. Diagrama de caja de tiempo en segundos sobre la cantidad de artículos de un pedido durante la prueba con sistema del experimento 3.	101
B.27. Diagrama de caja de tiempo en segundos sobre la cantidad de artículos distintos de un pedido durante la prueba con sistema del experimento 3.	102
B.28. Diagrama de caja del número de pedidos previamente en cola a la hora del ingreso de un nuevo pedido durante la prueba con sistema del experimento 3.	102
B.29. Errores ocurridos durante la prueba con sistema del experimento 3.	103
B.38. Gráfico de barras de número de pedidos durante la prueba sin sistema del experimento 3.	103
B.30. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 24/05/2023 del experimento 3.	104
B.31. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 25/05/2023 del experimento 3.	105
B.32. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 26/05/2023 del experimento 3.	106
B.33. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 27/05/2023 del experimento 3.	107
B.34. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 30/05/2023 del experimento 3.	108
B.35. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 31/05/2023 del experimento 3.	109
B.36. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 01/06/2023 del experimento 3.	110
B.37. Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 02/06/2023 del experimento 3.	111
B.39. Diagrama de caja de número de pedidos durante la prueba sin sistema del experimento 3.	112
B.40. Gráfico de barras de media de tiempo en segundo que demora un pedido durante la prueba sin sistema del experimento 3.	112
B.41. Diagrama de caja de tiempo en segundo que demora un pedido durante la prueba sin sistema del experimento 3.	112

B.42. Gráfico de barras de total de artículos por día durante la prueba sin sistema del experimento 3.	113
B.43. Gráfico de barras de media de artículos de un pedido por día durante la prueba sin sistema del experimento 3.	113
B.44. Diagrama de caja de número de artículos durante la prueba sin sistema del experimento 3.	113
B.45. Gráfico de barras de media de artículos distintos de un pedido durante la prueba sin sistema del experimento 3.	114
B.46. Diagrama de caja de media de artículos distintos de un pedido durante la prueba sin sistema del experimento 3.	114
B.47. Diagrama de caja de tiempo en segundos sobre la cantidad de artículos de un pedido durante la prueba sin sistema del experimento 3.	114
B.48. Diagrama de caja de tiempo en segundos sobre la cantidad de artículos distintos de un pedido durante la prueba sin sistema del experimento 3.	115
B.49. Diagrama de caja del número de pedidos previamente en cola a la hora del ingreso de un nuevo pedido durante la prueba sin sistema del experimento 3.	115
B.50. Errores ocurridos durante la prueba sin sistema del experimento 3.	116
B.51. Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 06/06/2023 del experimento 3.	117
B.52. Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 07/06/2023 del experimento 3.	118
B.53. Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 08/06/2023 del experimento 3.	119
B.54. Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 09/06/2023 del experimento 3.	120
B.55. Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 10/06/2023 del experimento 3.	121
B.56. Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 12/06/2023 del experimento 3.	122

Índice de tablas

3.1. Conjunto inicial de conceptos.	10
3.2. Conjunto inicial de reglas.	10
3.3. Conjunto final de conceptos.	11
3.4. Conjunto final de reglas.	11
3.5. Alimentos de prueba 1.	12
3.6. Pedidos de prueba 1.	12
3.7. Estaciones de trabajo de prueba 1.	12
3.8. Resultado esperado de prueba 1.	12
3.9. Prototipo de modelo de solución.	13
3.10. Modelo de solución del prototipo final.	29
4.1. Estaciones de trabajo para pruebas.	39
4.2. Artículos para pruebas.	39
4.3. Prioridades para pruebas.	40
4.4. Combinaciones para pruebas.	40
4.5. Prueba 1 (pedidos).	40
4.6. Prueba 1 (artículos).	40
4.7. Prueba 1 (resultado esperado).	40
4.8. Prueba 2 (pedidos).	41
4.9. Prueba 2 (artículos).	41
4.10. Prueba 2 (resultado esperado).	41
4.11. Prueba 3 (pedidos).	41
4.12. Prueba 3 (artículos).	41
4.13. Prueba 3 (resultado esperado).	42
4.14. Prueba 4 (pedidos).	42
4.15. Prueba 4 (artículos).	42
4.16. Prueba 4 (resultado esperado).	42
4.17. Prueba 5 (pedidos).	43
4.18. Prueba 5 (artículos).	43
4.19. Prueba 5 (resultado esperado).	43
4.20. Prueba 6 (pedidos).	43
4.21. Prueba 6 (artículos).	44
4.22. Prueba 6 (resultado esperado).	44
4.23. Prueba 7 (pedidos).	44
4.24. Prueba 7 (artículos).	44
4.25. Prueba 7 (resultado esperado).	45

4.26. Prueba 8 (pedidos).	45
4.27. Prueba 8 (artículos).	45
4.28. Prueba 8 (resultado esperado).	45
4.29. Hoja 1: “artículos”.	52
4.30. Hoja 2: “combinaciones”.	52
4.31. Hoja 3: “notas”.	52
5.1. Diferentes errores cometidos u observados en la experiencia y frecuencia de usuarios.	54
5.2. Tareas y las veces que fueron percibidas como difíciles.	55
5.3. Pruebas definidas para las vistas de pedidos y las veces que pasaron	56
5.4. Errores ocurridos en el restaurante con el sistema y las veces que pasaron	58
5.5. Errores ocurridos en el restaurante sin el sistema y las veces que pasaron	60

Capítulo 1

Introducción

En los servicios de restaurante pueden surgir muchos problemas que afectan la eficacia en la prestación de un buen servicio. Actualmente, algunos de estos problemas se solucionan mediante la tecnología, por ejemplo, la digitalización de los pedidos para evitar su extravío, digitalizar la carta para evitar posibles problemas al calcular los precios [3], entre otros.

Una de las tecnologías que se utilizan en la gestión de restaurantes son los TPV o terminales de punto de venta (o POS por sus siglas en inglés *Point Of Sale*)¹. Estos dispositivos permiten gestionar tareas relacionadas con la venta, y en la creación e impresión de tickets de venta, digitalizando la carta del establecimiento para agilizar la toma de pedidos. Existen distintos programas con variados precios y capacidades, como *Toast*², *Itactil*³, *Chromis*⁴ y *Square*⁵.

Estas herramientas poseen cierto nivel de integración con la cocina, pero no van más allá de mostrar los pedidos en pantalla una vez que se han ingresado al sistema. Esto es útil en la mayoría de los casos. Sin embargo, es común que durante la hora del almuerzo lleguen muchos pedidos a la vez, lo que dificulta decidir qué platos priorizar.

Algunos casos triviales al decidir qué platos priorizar, serían que lleguen clientes que quieran un mismo plato o en donde no coincidan en ninguno, pues la solución sería sacar los platos de los pedidos según la orden de llegada. Donde en el primer caso, como se quiere sacar los pedidos en orden de llegada, sería trivial, ya que, para completar los pedidos, se tendría que sacar la cantidad necesaria del plato.

Ahora bien, lo normal sería que llegasen pedidos en donde algunos platos coinciden y otros no. La solución óptima a este caso sería tomar el primer pedido, ver si algún plato del pedido es solicitado en otros pedidos; si es así, se debería preparar la cantidad necesaria, pues en caso contrario, se deberá preparar nuevamente el plato llegado el momento, demorándose más que si hubiera preparado una mayor cantidad en la primera ocasión. Por otro lado, si no hay más pedidos con el mismo plato, se prepara y se pasa al siguiente, pues en caso

¹<https://www.investopedia.com/terms/p/point-of-sale.asp> (5 de septiembre de 2022)

²<https://pos.toasttab.com> (5 de septiembre de 2022)

³<https://www.infotronic.es/software-tpv-bar-y-restaurant-hosteleria> (5 de septiembre de 2022)

⁴<https://chromis.co.uk> (5 de septiembre de 2022)

⁵<https://squareup.com/us/es> (5 de septiembre de 2022)

contrario, sobrarían los alimentos, implicando en pérdidas.

Además, si también se atienden pedidos para servir, uno como restaurante tiene un orden para ir sacando los platillos, cosa que se ha podido apreciar no sólo ayudando en un restaurante por más de diez años, sino que también al visitar diversos restaurantes de distintos países del mundo como *Tony Roma's*⁶ y *Niu Sushi*⁷ en Chile, *Red Lobster*⁸ en Estados Unidos y *Tosokchon Samgyetang*⁹ en Seúl, Corea. Dentro de estos establecimientos, se cumple con el protocolo de servir primero las entradas y luego los platillos principales. Esta particularidad, junto con los casos mencionados anteriormente, complicó aún más el problema de optimización que se resolvió.

Aterrizando más en detalle al problema a tratar en este trabajo de título, se dividirá en dos partes. La primera sería modelar el problema de optimización con la mayor cantidad de variables posibles. algunas de ellas serían las distintas estaciones de trabajo, pues no todos los alimentos proceden de una misma estación, la cantidad de estaciones y capacidad de cada una de ellas, los pedidos en sí mismos y la prioridad de los elementos que la conforman.

Por otro lado, la segunda parte sería diseñar e implementar un software que aplique esta lógica, en donde se hará uso algún tipo de dato abstracto, tal como las colas de prioridad, pues con las diversas variables ya mencionadas se podría calcular una prioridad. Aún no está definida la opción a tomar, se decidirá tras definir bien el modelo de solución de la primera parte.

Por los casos ejemplificados y los factores que afectan al problema de optimización, el uso de la tecnología serviría para solucionar el problema a abordar. Por otra parte, como ya existen programas de código abierto tales como *Square* u *Odoo POS*¹⁰ que permiten implementar una interfaz TPV para el ingreso de comandas, se decidió emplear alguno de ellos para centrarnos únicamente en el desarrollo de la solución como tal.

1.1. Objetivos

Objetivo General

Diseñar e implementar un sistema destinado a optimizar la preparación de alimentos en restaurantes. Este sistema indicará el orden adecuado para cocinar los platillos, basándose en el orden de llegada de los pedidos y en las capacidades específicas de cada restaurante. El objetivo es mejorar el tiempo que les toma a los restaurantes preparar sus pedidos y disminuir el número de consultas realizadas para determinar qué platillos y en qué cantidad deben cocinarse.

⁶<https://tonyromas.com> (10 de septiembre de 2022)

⁷<https://www.niusushi.cl> (10 de septiembre de 2022)

⁸<https://www.redlobster.com> (10 de septiembre de 2022)

⁹https://www.tripadvisor.com/Restaurant_Review-g294197-d1174982-Reviews-Tosokchon_Samgyetang-Seoul.html (10 de septiembre de 2022)

¹⁰https://www.odoo.com/es_ES (5 de septiembre de 2022)

Objetivos Específicos

Con el fin de poder lograr el objetivo general, se propuso seguir los siguientes objetivos específicos:

1. Diseñar un modelo de solución para el problema de optimización. Para comprobar su correctitud, se someterá a simulaciones de casos reales una vez confeccionado.
2. Diseñar e implementar un modelo de datos completo, que permita el manejo correcto del modelamiento del problema de optimización. Para comprobar la correctitud del modelo, se poblará con datos reales de un restaurante.
3. Diseñar e implementar un back-end que permita emplear de manera correcta un algoritmo que simule el problema de optimización. Para comprobar la validez del algoritmo, se probará mediante simulaciones de casos reales.
4. Diseñar e implementar una interfaz de usuario usable e intuitiva, donde se muestre claramente el orden de los platillos que se deben cocinar y en la que el personal pueda confirmando cada vez que se cocine un platillo, para avanzar al siguiente.
5. Implementar una interfaz de usuario usable para ingresar los pedidos al sistema, que sea lo más parecido posible a un sistema *TPV* convencional, en donde simplemente se requiera hacer un sistema de grillas con distintos grupos de alimentos, tal que al presionarlos en ellos se muestren todos los alimentos de dicho grupo.

Para tener una solución que sea aplicable a otros restaurantes, se decidió que la solución debe ser una que sea portable, que tenga una clara separación entre los datos y los algoritmos.

1.2. Solución Desarrollada

La solución desarrollada para el problema planteado en este trabajo de título tuvo como objetivo crear una aplicación que ayude en el proceso de decidir el orden de preparación de platillos dados ciertos parámetros.

El producto final desarrollado en este proyecto posee las siguientes características:

1. La aplicación permite al usuario ingresar de pedidos al sistema, en donde un pedido está conformado por los artículos, notas, datos del cliente, el tipo de pedido y si es programado o no.
2. La aplicación permite al usuario manejar los pedidos pendientes desde el interior de la cocina mediante una vista que muestra los pedidos pendientes en orden de llegada.
3. La aplicación permite al usuario visualizar la cola de cada estación de trabajo en forma de lista para saber el orden de los platillos a preparar.
4. La aplicación permite al usuario realizar un cierre de las ventas del día necesario en caso de querer un registro de estas mismas.

En cuanto a requisitos no funcionales:

1. Las interfaces de usuario tengan una baja curva de aprendizaje.
2. Las colas de las estaciones se muestran manera correcta.

Al contrastar las características del prototipo final con los objetivos específicos, se lograron cumplir los primeros tres objetivos sin mayores problemas. En cuanto al cuarto objetivo, se desarrolló una interfaz que muestra la información de pedidos y artículos en orden, aunque se reconoce que no resultó tan intuitiva como se tenía pensado en un inicio; sin embargo, la interfaz cuenta con una baja curva de aprendizaje. Por otro lado, el quinto objetivo no se alcanzó en su totalidad, ya que finalmente se optó por crear una interfaz TPV propia en lugar de implementar un sistema ya existente. Esto se hizo para tener una mayor libertad de creación. A pesar de esto, el prototipo final se considera como terminado.

1.3. Estructura del documento

Los próximos capítulos irán desarrollando las ideas, decisiones de diseño, desarrollo de la solución, evaluación y sus resultados, para finalmente hacer un análisis de estos y concluir. Los capítulos específicos, con su respectivo resumen, son:

El capítulo 2 se titula “Trabajo relacionado”. Lo primero que es abordado en este capítulo es el estado del arte de diversos sistemas relacionados con el trabajo desarrollado en la memoria. Estos sistemas se encontraron buscando palabras claves dentro del motor de búsqueda de *Google*, llegando así a encontrar diversos sistemas en el mercado, describiendo en el mismo capítulo cómo se realizó la búsqueda, los sistemas y aplicaciones encontradas mediante esta misma y las diferentes funcionalidades que ofrecen cada una, además de mencionar el método empleado para resolver el problema de optimización.

El capítulo 3 se titula “Desarrollo de la solución”. En este apartado de la memoria se aborda el desarrollo completo del prototipo, desde la selección del framework de trabajo, la definición del conjunto de reglas para la solución en colaboración con expertos, el prototipo inicial y final del modelo de solución para el problema de optimización. Además, se describen los pasos seguidos para desarrollar el prototipo final del sistema. Dicho prototipo final consta de tres vistas principales: una para el ingreso de pedidos o vista TPV (terminal de punto de venta) y dos vistas para mostrar el detalle de los pedidos y colas de preparación de alimentos de distintas estaciones de trabajo.

El capítulo 4 se titula “Evaluación de la solución”. En él se definen tres protocolos para evaluar distintos aspectos del prototipo final. Uno de ellos, para la evaluación del sistema de toma de pedidos con usuarios reales. Otro para la evaluación del comportamiento del sistema bajo un ambiente de laboratorio, definiendo tareas con resultados esperados basándonos en la experiencia de expertos dentro del dominio. Por último, una evaluación en un ambiente real dentro de un restaurante de comida china para servir y llevar.

El primer experimento corresponde a una prueba en el cual se evaluaron aspectos cuantitativos. Para ello, se propusieron tareas que ejecutar en la aplicación, por ejemplo, ingresar pedidos en el sistema, midiendo algunas de las métricas de usabilidad de Wilson y Wixon [4], las cuales son el tiempo en completar una tarea, número y tipo de errores por tarea, número de errores por unidad de tiempo, número de ayudas necesarias y número de usuarios que completan una tarea con éxito, de las cuales sólo se midieron el número de errores por tarea, número de ayudas necesarias y número de usuarios que completan una tarea con éxito, pues el desarrollo de esta vista no es la parte más esencial que se quiere resolver en este trabajo

de título. Este proceso se realizó con ayuda de usuarios reclutados por el memorista para realizar las tareas mencionadas anteriormente. Tras la experiencia, se les hizo una encuesta para descubrir qué partes de las tareas se percibieron como las más complejas por los usuarios y así detectar posibles mejoras dentro del sistema.

El segundo experimento corresponde a completar una serie de pruebas con un resultado esperado basado en la experiencia de expertos, donde cada prueba corresponde a una serie de acciones para llegar a un resultado esperado, teniendo en cuenta un contexto bien definido, pues el problema de optimización depende totalmente del contexto en el cual se define. Cada prueba se realizó múltiples veces para tener una mayor certeza de que el funcionamiento del sistema es el esperado que no haya pasado o reprobado una prueba por algún error.

Por último, el tercer experimento corresponde en la puesta a prueba del sistema en un restaurante de comida china real y someterlo a casos de usos diarios durante unos días para recolectar información de errores que surgieron mientras se ocupaba la aplicación y algunas métricas acerca del tiempo que se demoran los pedidos, seguido de una segunda fase del experimento en donde se recopilaran las mismas métricas, además del registro de errores en el mismo establecimiento para tener un conjunto de datos con el que comparar.

Todos los instrumentos, métricas y métodos empleados para cada experimento se encuentran descritos en mayor profundidad dentro de este capítulo.

El capítulo 5 se titula “Resultados”. En este apartado de la memoria se muestran los resultados de la etapa de evaluación. Para eso se presentaron los resultados de los 3 experimentos en secciones del capítulo. En la primera sección se presentan los resultados del experimento realizado con usuarios que utilizaron la vista TPV para seguir una serie de tareas, en donde se resume las métricas de Wilson y Wixon [4] obtenidas durante la experimentación, seguido del resultado de la encuesta que se les tomó tras la experiencia para ver aspectos a mejorar según lo percibido por los usuarios. En la segunda sección se presentan los resultados del segundo experimento, indicando cuantas veces se pasó una prueba tras ser repetida 5 veces cada una. Por último, para los resultados del tercer experimento, primero se presentan errores del sistema encontrados durante el período en el que se probó el sistema en el restaurante, y el tiempo que se demoró en preparar un pedido desde su ingreso. Seguidos de los resultados de las mismas métricas en el mismo restaurante sin la aplicación. Los detalles de cada experimento se encuentran en el capítulo.

El capítulo 6 se titula “Análisis y discusión de resultados”. En este apartado se presentan los análisis y supuestos obtenidos a partir de los resultados. Al analizar los datos del primer experimento, la principal conclusión obtenida fue que el flujo de toma de pedidos no es muy intuitivo en su estado actual, aunque sí posee una baja curva de aprendizaje. Además, se generaron propuestas de vistas basadas en los errores identificados, combinados con los comentarios dejados por los usuarios después de la experiencia.

Por otro lado, la segunda prueba no es concluyente, ya que es meramente una prueba preliminar para verificar que el sistema se comporte de la manera esperada antes de enfrentarlo a casos reales.

En cambio, de los resultados de la tercera prueba se pudo concluir que el sistema sí aporta en la cocina del restaurante. Al igual que con los resultados de la primera prueba, se emplearon los errores identificados para diseñar propuestas de vistas que resolvieran estos problemas.

Por último, en este mismo capítulo se discuten posibles extensiones que se le pueden aplicar al sistema.

El capítulo 7 se titula “Conclusión y Trabajo Futuro”. En este apartado se documentan las reflexiones obtenidas del proceso, tanto de los aprendizajes y de los resultados. Se identifican también las oportunidades de mejora para ir agregándolas en futuras iteraciones. Se hace alusión también al producto obtenido de esta memoria. Se espera que, en la siguiente iteración del desarrollo, se agreguen oportunidades de mejora con criticidad alta, para probar el sistema en diferentes restaurantes y durante un tiempo más prolongado, para asegurar si el sistema desarrollado apoya en los negocios de este rubro, algo que quedó parcialmente cumplido en el desarrollo de esta memoria.

Capítulo 2

Trabajo relacionado

2.1. Aplicaciones existentes

Antes de poder diseñar la solución, se tuvo que revisar la situación actual de las herramientas que ayudan a mitigar posibles errores u optimizaciones que pueden surgir en una cocina de un restaurante.

Tras realizar una investigación preliminar en el motor de búsqueda *Google*, utilizando frases clave en inglés como “*Kitchen management software*”, “*POS apps with kitchen integration*” y “*Kitchen planners*”, cuyas traducciones correspondientes al español son “*Software de gestión de cocinas*”, “*Aplicaciones TPV con integración en la cocina*” y “*Planificadores de cocina*”, se logró identificar una variedad de aplicaciones especializadas en la gestión de cocinas.

Cabe mencionar que se listan las herramientas que más se diferencian entre sí, por más que existan otras opciones. La presentación de estas sirvió para tomar cierta inspiración, y también para identificar oportunidades de mejora. Una vez dicho esto, se procede a enumerar las herramientas.

- Jamix¹: Aplicación que permite el cálculo de los ingredientes necesarios para cocinar ciertas porciones de comida y da la opción de comprarlos en línea. Es complejo de ingresar menús en grandes cantidades, debido a que se deben ingresar uno por uno.
- Toast²: Aplicación TPV que muestra los pedidos en la cocina. A simple vista se ve que es muy buena, debido al amplio abanico de integraciones que ofrece, pero no posee lo que se quiere desarrollar, que es el mostrar un orden de los platillos a elaborar.
- Xtrachef³: Aplicación que ayuda al manejo de inventario hecho por la empresa *Toast*. Dice que ayuda al caos en la cocina; sin embargo, no especifica de qué manera lo realiza. Usado por la cadena Papa Johns.
- Apicbase⁴: Aplicación que ayuda a la cocina haciendo una planificación con los platillos

¹<https://www.jamix.com> (5 de septiembre de 2022)

²<https://pos.toasttab.com> (5 de septiembre de 2022)

³<https://xtrachef.com> (5 de septiembre de 2022)

⁴<https://get.apicbase.com> (5 de septiembre de 2022)

que ya están, los que se están haciendo y los que quedan por cocinar. Se parece a lo que se está buscando desarrollar, pero no lo es exactamente, pues uno tiene que ingresar el total de los platillos que se tienen que cocinar a mano. Lo ideal sería que se pueda ingresar pedidos por separado y que el programa entregue un orden de manera óptima de los platillos que se tengan que ir cocinando según orden de llegada y capacidad de cada línea de trabajo dentro de la cocina, para así minimizar el tiempo de espera de los clientes.

- Posist⁵: Aplicación TPV, la cual muestra los pedidos en una pantalla dentro de la cocina, pero tampoco cuenta con la optimización que se desea implementar.
- Oracle MICROS POS⁶: Aplicación TPV, la cual muestra los pedidos en una pantalla dentro de la cocina. Tiene un gran ecosistema, por la variedad de productos y opciones que ofrece. Además de ofrecer hardware de TPV a \$1⁷, lo que incentiva a pequeñas empresas a adquirirlos por su bajo costo, pero tampoco resuelve el problema abordado, pues no optimiza el orden en el que deberían salir los platillos para reducir los tiempos de preparación de los alimentos.
- NewPOS⁸: Aplicación TPV online, la que muestra los pedidos en una pantalla dentro de la cocina; sin embargo, tampoco posee la capacidad de mostrar los platillos a cocinar a los cocineros, que es lo que se quiere integrar. Cuenta con tecnología muy avanzada para grandes negocios, como para recibir cientos de miles de pedidos al día, cosa que sería demasiado para una mediana o pequeña empresa. Según la búsqueda de las palabras claves *"Which POS system does Mcdonald's use"* lo que se traduce del inglés a *"Que sistema TPV usa Mcdonald's"* en Google, Mcdonald's usó esta aplicación como base para implementar su propio sistema TPV.

Gran parte de las aplicaciones mencionadas ofrecen mucha ayuda a los negocios, pero al momento de hacer la investigación, ninguna de estas resolvía el problema que se quiere abordar. Esto es porque la mayoría poseen un nivel de integración que solo muestra los pedidos en una pantalla dentro de la cocina y les faltaría que mostrasen un orden que indique qué platillos se preparen.

2.2. Estado de la práctica

Como el problema de optimización a resolver no está definido como tal, pues varía de restaurante en restaurante, el problema se abordó mediante una metodología a base de heurísticas [2]. Razón por la cual, se optó por definir un conjunto de reglas tanto para definir el problema en particular como para su resolución, en vez resolver el problema mediante un algoritmo más complejo.

⁵<https://www.posist.com/cloud-kitchen-software/> (5 de septiembre de 2022)

⁶<https://www.oracle.com/industries/micros/> (5 de septiembre de 2022)

⁷<https://www.oracle.com/industries/food-beverage/restaurant-pos-systems/pos-hardware/> (5 de septiembre de 2022)

⁸<https://www.newpostech.com/en/index.aspx> (5 de septiembre de 2022)

Capítulo 3

Desarrollo de la solución

En este capítulo, se describen de manera secuencial los pasos seguidos para llegar al prototipo final utilizado en la fase de evaluación.

3.1. Análisis y Modelado

3.1.1. Variables del Modelo de Optimización

Inicialmente, se identificaron diversos conceptos y reglas que podrían tener relevancia para el negocio, basándose en observaciones directas del funcionamiento del negocio en el que se implementará el software.

A través de un proceso iterativo, validamos dichos conjuntos de reglas y conceptos con la colaboración de cinco expertos en el área. Estos expertos, con experiencia en diferentes restaurantes de comida china y con quienes teníamos algún grado de familiaridad, aceptaron amablemente nuestra invitación para entrevistarlos.

El objetivo central es garantizar una definición integral de las bases del modelo de optimización. Esto permite reducir la posibilidad de enfrentar problemas con las variables al momento de desarrollar el modelo, disminuyendo así los cambios necesarios durante el desarrollo del software.

A continuación, se detalla el conjunto inicial de conceptos y reglas que se presentó a los expertos:

Tabla 3.1: Conjunto inicial de conceptos.

Concepto	Descripción
1.Pedido	Lista de alimentos solicitados por un cliente y tiempo de ingreso al sistema.
2.Pedido programado	Casos en donde un pedido es para una hora en específico.
3.Tipo de pedido	El tipo que puede llegar a ser un pedido, por ejemplo, para servir o para llevar.
4.Alimento	Todos los elementos del menú de un restaurante, el nombre, la estación de trabajo donde se prepara y en caso de ser una combinación de otros alimentos, la lista de ellos.
5.Estación de trabajo	Lugar donde se prepara cierto tipo de alimento.

Tabla 3.2: Conjunto inicial de reglas.

Reglas
1.Preparar los alimentos de los pedidos para servir en un orden en específico.
2.Priorizar la preparación de alimentos según orden de llegada.
3.A la hora de preparar un alimento, tener en cuenta la capacidad máxima de la estación de trabajo y la cantidad total solicitada por los clientes, y preparar el mínimo entre las dos.
4.Priorizar aperitivos para pedidos para servir a pedidos para llevar.

Después de definir los conjuntos iniciales, el primer experto consultado, con más de 30 años de experiencia en restaurantes de comida china, corroboró la completitud de los conjuntos.

Al consultar al segundo experto, con más de 10 años en el sector y con experiencia tanto en cocina como en atención al cliente, destacó que en su restaurante de mayor demanda, durante las horas pico, preparan aperitivos en alta demanda, sin considerar posibles excedentes, para agilizar el servicio. Esta práctica difiere de la primera regla inicial.

El tercer experto, cuya experiencia se basa en trabajar en un restaurante de comida exclusivamente para llevar por unos 10 años, coincidió en que los conceptos y reglas propuestas eran adecuados.

El cuarto experto propuso una adición importante: tener en cuenta los alimentos de mayor demanda, manteniendo una pequeña reserva preparada, especialmente durante las horas pico. Aseguró que esto podría agilizar el servicio sin incurrir en pérdidas significativas para el negocio.

Finalmente, la quinta experta, especialista en la organización de preparación y empaque de alimentos, corroboró la validez de los conjuntos y apoyó la sugerencia del cuarto experto.

Considerando las aportaciones de los expertos, se redefinieron los conjuntos de conceptos y reglas como los siguientes:

Tabla 3.3: Conjunto final de conceptos.

Concepto	Descripción
1.Pedido	Lista de alimentos solicitados por un cliente y tiempo de ingreso al sistema.
2.Pedido programado	Casos en donde un pedido es para una hora en específico.
3.Tipo de pedido	El tipo que puede llegar a ser un pedido, por ejemplo, para servir o para llevar.
4.Alimento	Todos los elementos del menú de un restaurante, el nombre, la estación de trabajo donde se prepara y en caso de ser una combinación de otros alimentos, la lista de ellos.
5.Estación de trabajo	Lugar donde se prepara cierto tipo de alimento.
6.Hora punta	Período de tiempo en donde hay un mayor número de pedidos.
7.Alimentos más pedidos	Conjunto de alimentos más pedidos.

Tabla 3.4: Conjunto final de reglas.

Reglas
1.Preparar los alimentos de los pedidos para servir en un orden en específico.
2.Priorizar la preparación de alimentos según orden de llegada.
3.A la hora de preparar un alimento, tener en cuenta la capacidad máxima de la estación de trabajo y la cantidad total solicitada por los clientes, y preparar el mínimo entre las dos.
4.Priorizar aperitivos para pedidos para servir a pedidos para llevar.
5.Si es horario punta y llega un pedido con un alimento que es una muy demanda, se comprueba si ya hay preparado, si no, se prepara más de lo solicitado.

Los conceptos 6 y 7, así como la regla 5, se añadieron basándose en las recomendaciones del cuarto experto. Si no se desean utilizar, simplemente pueden definirse como conjuntos vacíos.

3.1.2. Prototipo inicial de Modelo de solución

Para llevar a cabo esta tarea, comenzamos diseñando una batería de pruebas que nos aseguraran la correcta funcionalidad del modelo de solución propuesto. Estas pruebas consisten en casos reales que varían desde una simple orden de un alimento hasta situaciones más complejas que involucran distintos pedidos de menús compuestos por múltiples alimentos y con un orden de llegada específico.

Las pruebas se estructuran en cuatro tablas: una para los alimentos ingresados al sistema, una para los pedidos registrados, una tercera para definir las características de las estaciones de trabajo, y una última para mostrar el orden de cocción esperado. Algunas de estas pruebas representan casos reales extraídos del restaurante donde se realizarán las evaluaciones, un ejemplo sería el siguiente:

Tabla 3.5: Alimentos de prueba 1.

Cantidad	Alimento	Pedido	Tipo de estación
1	Carne Mongoliana	1	1
4	Carne Mongoliana	2	1

Tabla 3.6: Pedidos de prueba 1.

Pedido	Orden	Tipo
1	1	Para llevar
2	2	Para llevar

Tabla 3.7: Estaciones de trabajo de prueba 1.

Estación	Tipo	Capacidad Máxima
Platillos_01	1	4
Frituras_01	2	3

Tabla 3.8: Resultado esperado de prueba 1.

Estación	Alimento	Cantidad
Platillos_01	Carne Mongoliana	1
Platillos_01	Carne Mongoliana	4

Una vez conformada nuestra batería de pruebas, iniciamos el diseño de un prototipo del modelo de solución para el problema de optimización. Este modelo está compuesto por un conjunto de reglas que buscan resolverlo de manera efectiva. El prototipo se detalla en la siguiente lista de reglas:

Tabla 3.9: Prototipo de modelo de solución.

Reglas
1. Encontrar el primer pedido p de la lista de pedidos lp que no ha sido encolado, pasar a 2. Si no existe p , terminar.
2. Encontrar el primer elemento e del pedido p que aún no esté encolado, si existe e , pasar a 3. Si no, marcar p como encolado y devolverse a 1.
3. Verificar sobras del mismo tipo de alimento t de e , si hay suficiente cantidad cs de las sobras, restar la cantidad c de e a cs , en caso contrario, restar la cs a c , pasar a 4.
4. Encontrar la lista de estaciones de trabajo let que sea del tipo de estación de trabajo de e , pasar a 5.
5. Calcular (método por definir) la prioridad total pt de e , pasar a 6.
6. Encontrar estaciones de trabajo est de let que tenga un elemento $e2$ de tipo t , si hay alguno con una cantidad $c2$ inferior a la capacidad máxima max de la estación de trabajo, establecerla como estación de trabajo et , pasar a 7. Si no, establecer et como la estación de let con menor cantidad de elementos encolados, crear $e2$ con de tipo t , encolar $e2$ a et , pasar a 7.
7. Establecer la prioridad pr de $e2$ como el máximo entre pr y pt , pasar a 8.
8. Verificar si $max - c2 \geq c$, si es así, pasar a 9. Si no, establecer $c2$ como max y c como $c + c2 - max$, pasar a 11.
9. Verificar si es hora punta, si es así, pasar a 10. Si no, pasar a 11.
10. Verificar si t es muy demandado, si es así, establecer $c2$ como max y pasar a 12. Si no, pasar a 11.
11. Establecer $c2$ como $c2 + c$, pasar a 12.
12. Marcar e como encolado y pasar a 13.
13. Devolverse a 2.

La validación de este modelo se llevará a cabo durante las primeras semanas del siguiente periodo, utilizando la batería de pruebas y simulando situaciones reales.

3.1.3. Exploración de tecnologías

Considerando la infraestructura del restaurante donde se efectuarán las pruebas, determinamos que lo más conveniente es desarrollar una aplicación web. Esto se debe a que se requiere una conexión entre la caja y la cocina. Una solución vía wifi resulta más viable que una conexión física mediante cables.

Respecto a los dispositivos que se usarán, hemos optado por un equipo que ejecute el software en la caja y una tableta con acceso a internet para la cocina. Esto porque, en la caja, hay suficiente espacio para albergar un dispositivo más grande, mientras que, en la cocina, debido a las restricciones de espacio, es necesario un dispositivo compacto, pero que aún permita una fácil lectura de las órdenes.

Finalmente, decidimos desarrollar el sistema en *Python*, utilizando el framework *Django*, motivado por las razones expuestas en la sección Solución Propuesta 1.2.

Para poner en marcha un servidor en la red wifi local, simplemente es necesario ejecutarlo en la dirección IP 0.0.0.0 y el puerto 8000 al correr el servidor de Django (utilizando el comando: *Python manage.py runserver 0.0.0.0:8000*). Después, hay que incluir la dirección IP

del dispositivo servidor en el campo `ALLOWED_HOSTS` del archivo `settings.py` del proyecto. Para acceder a la aplicación web desde otro dispositivo, simplemente hay que conectarse a través de la dirección IP configurada y el puerto 8000 desde un equipo que esté en la misma red wifi.

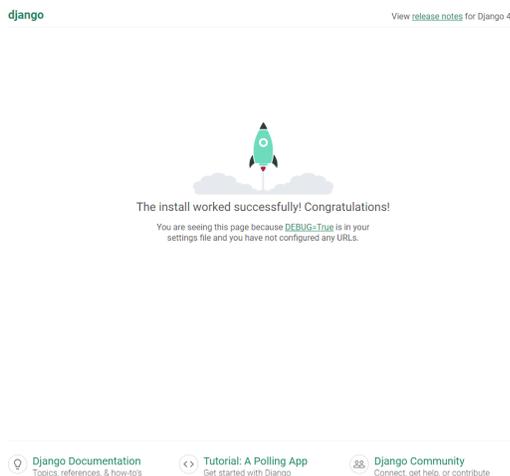


Figura 3.1: Página por defecto de *Django* desde dispositivo ejecutor.

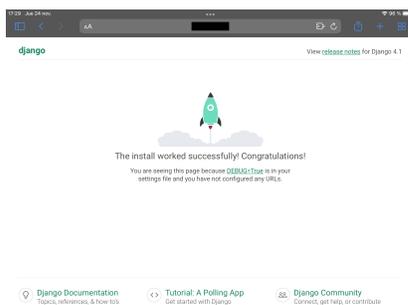


Figura 3.2: Página por defecto de *Django* desde dispositivo secundario.

3.1.4. Modelo de datos inicial

Con el objetivo de diseñar un software adecuado para un restaurante específico, se desarrolló un modelo de datos basándonos en el funcionamiento interno del mismo restaurante donde se implementará el sistema. Para este modelo, se consideraron tanto los elementos físicos como los conceptuales inherentes al funcionamiento del restaurante. Estos elementos se describen con más detalle en la sección “Variables del modelo de optimización” 3.1.1.

A continuación, se presenta el modelo propuesto:

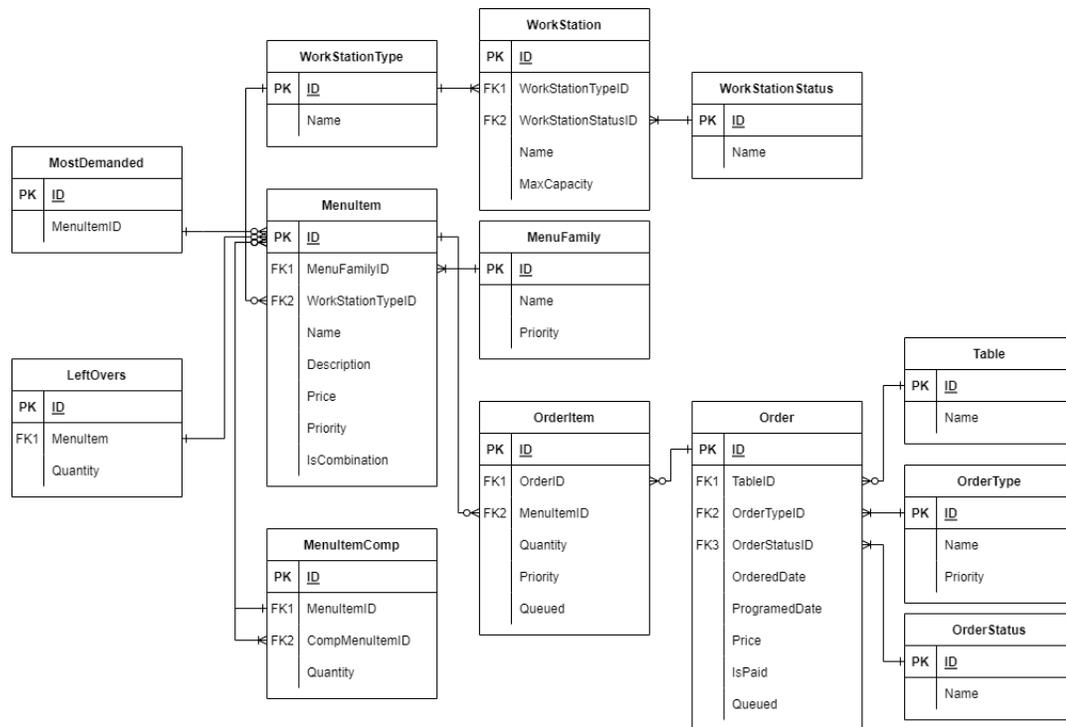


Figura 3.3: Modelo de datos inicial.

Dentro del modelo, optamos por implementar entidades que representan tanto objetos físicos — como las mesas (*Table*) y los alimentos (*MenuItem*) — como conceptuales, que abarcan listas de platillos más solicitados (*MostDemanded*), estaciones de trabajo (*WorkStation*), categorías de menús (*MenuFamily*) y pedidos (*Order*).

Se incluyeron también entidades más específicas para abordar necesidades particulares. Un ejemplo es *MenuItemComp*, que se refiere a elementos dentro de *MenuItem* que son combinaciones de otros elementos. Por ejemplo, un “Menú para 2” podría consistir en dos porciones de “Arroz” y una porción de “Carne”. La intención es que esta entidad represente dichas relaciones.

Se anticipa que la prioridad de un artículo estará determinada por varios factores: la demanda del platillo (los platillos más solicitados tendrían mayor prioridad), la naturaleza del pedido (si es para llevar o para consumo en el establecimiento) y la categoría a la que pertenece el artículo (por ejemplo, los aperitivos podrían tener mayor prioridad en pedidos para consumo en sitio). Debido a esta complejidad, se sugiere que varias entidades tengan un atributo de prioridad, el cual será crucial para determinar la prioridad final de un elemento cuando se asigne a una estación de trabajo.

Adicionalmente, algunas tablas actúan como entidades de estado y tipo, representando los diversos estados y tipos que pueden adoptar otras entidades del modelo. Finalmente, la entidad *OrderItem* se refiere a una instancia específica de un artículo, lo que se distingue de *MenuItem*, que representa la idea general de un alimento.

3.2. Configuración del proyecto inicial

Al inicio del proyecto, se estableció un ambiente de trabajo utilizando *Django*. Para garantizar una navegación intuitiva, se optó por una plantilla minimalista con una barra de navegación lateral y otra superior. La plantilla seleccionada fue *AdminLTE 3*¹.

3.3. Diseño e implementación de sistema de toma de pedidos

En las primeras etapas, se tomó la decisión de desarrollar un sistema de ingreso de pedidos propio. A pesar de haber explorado varias opciones de código abierto, ninguna satisfizo completamente las necesidades del restaurante objetivo. Esta decisión permitió mayor libertad de desarrollo, aunque significó un trabajo adicional en comparación con la integración de un sistema preexistente.

Aunque originalmente se había planeado implementar este sistema más adelante, se consideró beneficioso desarrollar primero una interfaz para ingreso de pedidos, facilitando así las pruebas para el resto de la aplicación.

El desarrollo se dividió en tres fases: front-end, back-end y su interconexión.

Front-end

El diseño inicial se inspiró en la aplicación *Itactil*, utilizada en el restaurante objetivo y en otros establecimientos consultados.



Figura 3.4: Vista de ingreso de artículos (Itactil). Figura 3.5: Vista de total de pedido (Itactil).

De las cuales se tomaron las grillas de familias y artículos, la tabla de visualización de artículos y la estructura en general de la figura 3.4 para desarrollar la vista de para la aplicación:

¹<https://adminlte.io/themes/v3/> (18 de junio de 2023)

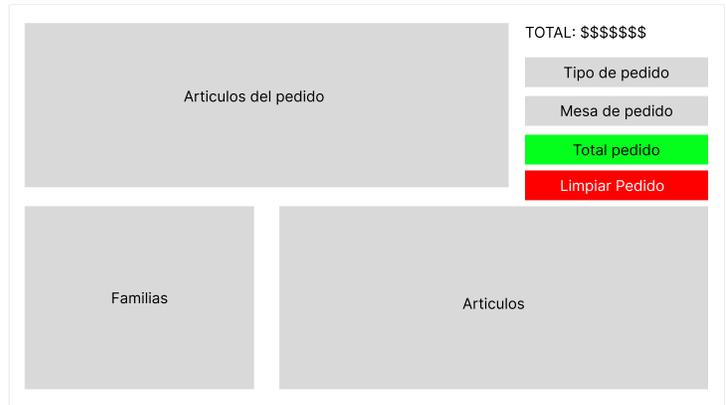


Figura 3.6: Mock-up de sistema TPV propio realizado en Figma.

La vista de la figura 3.5, aparece una vez se presiona el botón “Total” dentro de la aplicación Itactil, sin embargo, para la aplicación desarrollada se decidió ingresar opcionalmente los datos del cliente y mostrando un resumen de los artículos del pedido para una comprobación previa para ver que los datos sean los correctos.

Tras varias iteraciones y considerando las mejores prácticas de desarrollo móvil de *Android*², se refinó el diseño cambiando el scroll vertical a un scroll horizontal para obtener una interfaz intuitiva y eficiente.

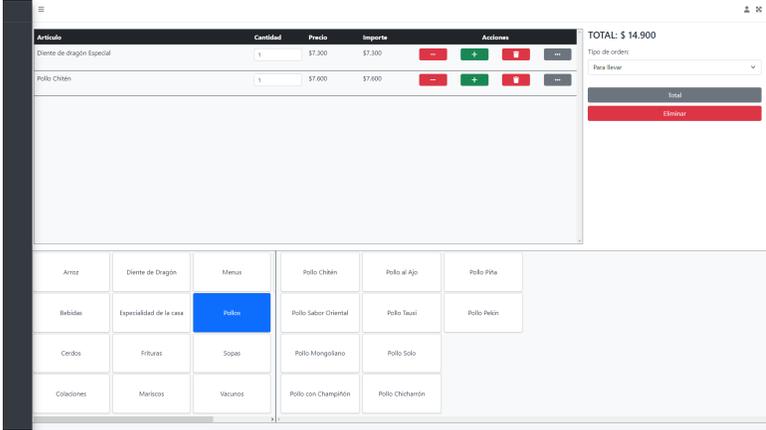


Figura 3.7: Vista prototipo final sistema TPV de la aplicación.

²<https://developer.android.com/develop/ui> (06 de julio de 2023)

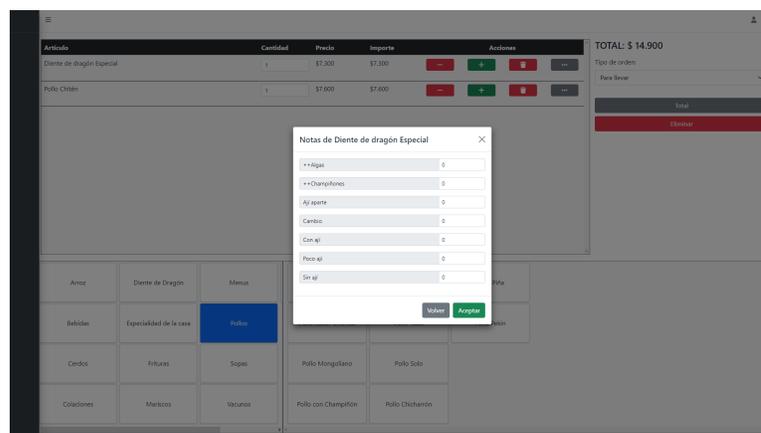


Figura 3.8: Vista prototipo final de notas de un artículo en el sistema TPV de la aplicación.

Vistas de las cuales se puede apreciar que la estructura de la aplicación final posee un sistema de familias con un scroll horizontal (sección de ubicada en la esquina inferior izquierda) y a su lado un sector para los respectivos artículos que pertenezcan a la familia seleccionada. Notar que los artículos se van llenando de izquierda a derecha de arriba hacia abajo, para que una vez seleccionada una familia el traslado del cursor sea el menor posible.

Back-end

En lo que respecta al back-end del sistema, se empezó por implementar una estructura de datos utilizando las sesiones de *Django*³. Esta implementación es crucial para almacenar temporalmente los pedidos y únicamente persistirlos en la base de datos cuando sea estrictamente necesario. Utilizando sesiones en lugar de variables globales, se pueden gestionar pedidos desde diferentes dispositivos, siempre que estén asociados a diferentes usuarios.

Un pedido, antes de ser registrado, se representa como una lista donde se insertan los artículos. Este pedido contiene distintas variables que abarcan desde el tipo de pedido, pasando por la mesa del pedido (si es para servir en mesa) y hasta la fecha programada del pedido (si este fuera el caso).

Al decidir guardar un pedido, el sistema procesa las variables del mismo almacenadas en la sesión para posteriormente crear los correspondientes objetos en la base de datos. En particular, se genera un objeto “Orden”, a cuyas propiedades se les asignan todas las variables, excepto la lista de artículos. Para esta última, por cada artículo en la lista, se crea un objeto “Orden_Artículo”, indicando la cantidad adecuada de dicho artículo.

Adicionalmente, se incorporó una función para importar automáticamente todo el menú a través de un archivo *Google Sheets*⁴. Esta elección se basó en la flexibilidad que *Google Sheets* brinda para ediciones y en la posibilidad de modificar el menú directamente desde la hoja de cálculo sin descargas adicionales. Para esta tarea, se utilizó la *API*⁵ proporcionada por

³<https://docs.djangoproject.com/en/4.2/topics/http/sessions/> (18 de junio de 2023)

⁴<https://www.google.com/sheets/about/> (18 de junio de 2023)

⁵<https://developers.google.com/sheets/api/quickstart/python?hl=es-419> (18 de junio de 2023)

Google, y siguiendo las instrucciones detalladas en su sitio web, la implementación se llevó a cabo sin inconvenientes significativos.

Conexión Front-end y Back-end

Con el front-end y el back-end ya desarrollados, el siguiente paso consiste en establecer una conexión entre ambos. Esta interacción se facilita a través del sistema de vistas proporcionado por *Django*. Adicionalmente, cualquier modificación en la interfaz requiere una actualización del pedido temporal en el back-end. Por ejemplo, al seleccionar un nuevo artículo, es necesario comunicarlo al back-end para añadir dicho elemento.

Para establecer esta conexión, se empleó la función *AJAX*⁶ de la API *jQuery*⁷. Desde el front-end, se realiza una petición al back-end utilizando solicitudes dirigidas a vistas diseñadas para tareas específicas. Retomando el ejemplo previo, al seleccionar un artículo para agregar, se dispara una función desde el front-end que envía una solicitud a una URL específicamente creada para añadir, basado en su ID, el artículo al pedido temporal.

Al concluir la operación en el back-end, esta devuelve una respuesta en formato *JsonResponse* de la biblioteca *Django.HTTP*. Dicha respuesta es procesada por la función que originó la solicitud para actualizar cualquier cambio reflejado en la interfaz.

3.4. Diseño e implementación de visualización de pedidos

Tras implementar una forma de ingresar pedidos, el siguiente paso es proporcionar una interfaz visual para que aquellos encargados de preparar los pedidos en la cocina puedan gestionarlos. Esta fase de desarrollo se describe nuevamente en las tres etapas de la sección anterior.

Front-end

Inicialmente, se diseñó una vista emulando tickets de cocina. Cada ticket muestra una lista de artículos, ordenados por el momento en que se recibieron, manteniendo la interfaz sencilla.

⁶<https://api.jquery.com/jquery.ajax/> (18 de junio de 2023)

⁷<https://api.jquery.com/> (18 de junio de 2023)

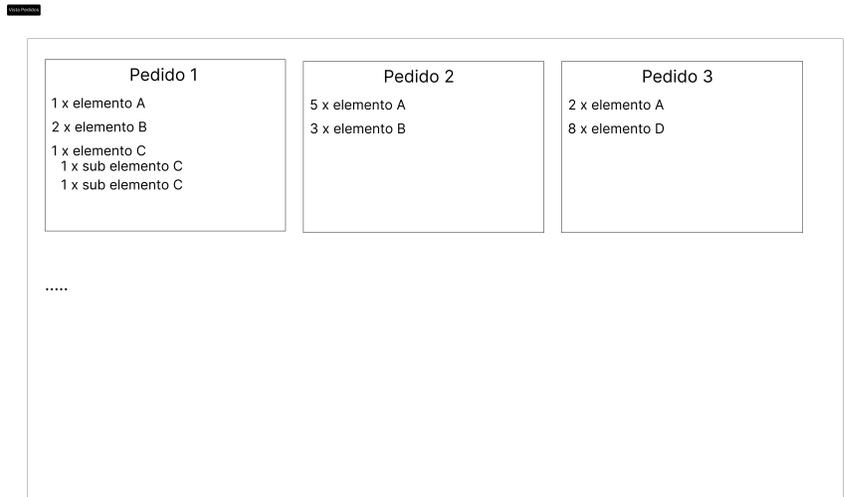


Figura 3.9: Mock-up 1 de *display* de pedidos realizado en Figma.

Tras el primer diseño, se decidió que los encargados de armar los pedidos también interactuarían con la aplicación para indicarle a la aplicación cuando algún artículo estuviera listo. Para ello se iteró sobre el diseño anterior 3.9 para generar la siguiente vista:

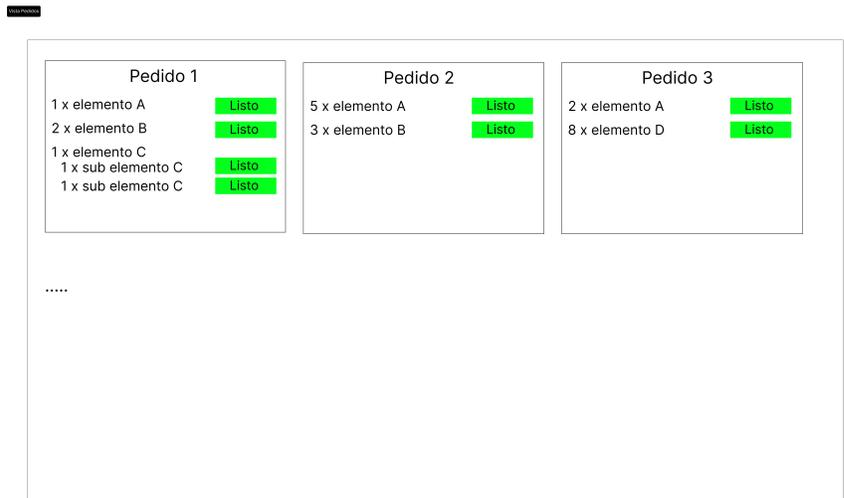


Figura 3.10: Mock-up 2 de *display* de pedidos realizado en Figma.



Figura 3.11: Mock-up 2 de *display* de pedidos tras presionar un botón de “Listo” realizado en Figma.

En dicha vista se colocó los botones de “Listo” a un lado de los artículos para que los encargados dentro de la cocina los puedan presionar e ingresar la cantidad de platos como se muestra en la figura 3.11.

Por último, para indicar el estado de un artículo se idearon dos principales métodos para ello, la primera es emplear el color de las fuentes para indicar el estado y una segunda forma en donde esté el estado a un costado de los artículos; sin embargo, se descartó esta última debido a que no se quería sobrecargar la vista.

Con base en los diseños, se obtuvieron las siguientes vistas del prototipo final:

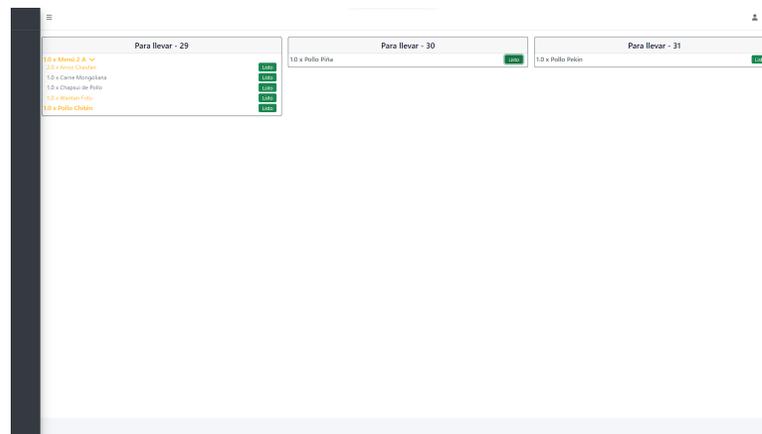


Figura 3.12: Vista prototipo final de pedidos pendientes de la aplicación.

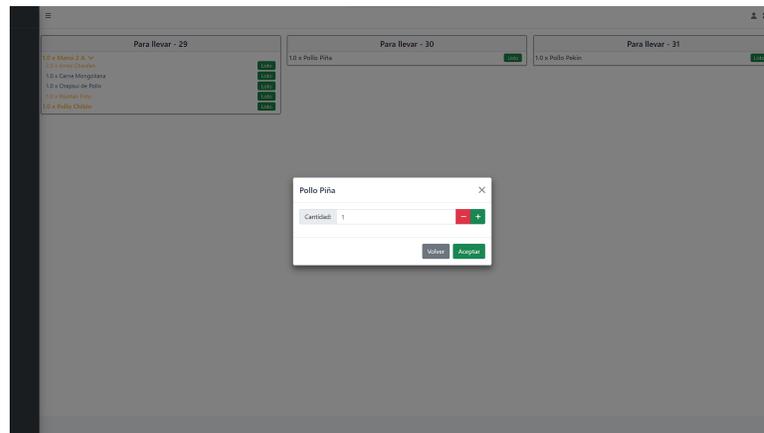


Figura 3.13: Vista prototipo final de pedidos pendientes (modal cantidad de artículos tras presionar botón “Listo”) de la aplicación.

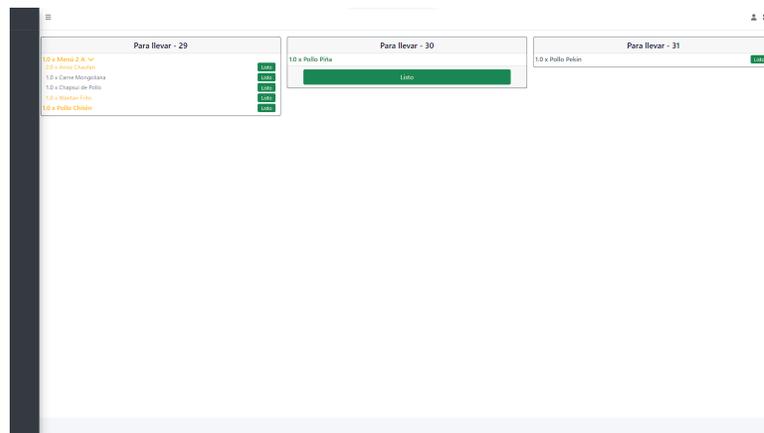


Figura 3.14: Vista prototipo final de pedidos pendientes (con un artículo listo) de la aplicación.

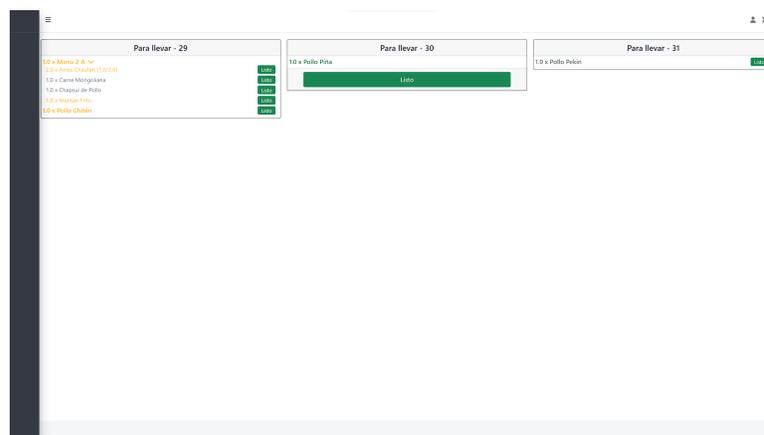


Figura 3.15: Vista prototipo final de pedidos pendientes (con un artículo parcialmente listos) de la aplicación.

Como se observa en las figuras 3.12 a 3.15, los colores simbolizan el estado de cada artículo: gris para aquellos pendientes, mostaza para los en proceso y verde para los ya listos.

El prominente botón de “Listo” en el pedido 30 de la figura 3.14 sólo se muestra cuando todos los artículos de un pedido están listos. Artículos compuestos por otros, como el “Menú 2A” del pedido 29, son elementos plegables que cambian a verde sólo cuando todos sus componentes están listos. Si un artículo está parcialmente listo, se muestra una fracción al lado indicando la cantidad preparada respecto a la solicitada.

Back-end

En esta etapa, se implementaron funciones para administrar las actualizaciones en la base de datos, reflejando la cantidad preparada de un artículo y el estado de un pedido.

Adicionalmente, si se registra una cantidad mayor a la solicitada para un artículo, esta diferencia se registra en la tabla “Leftovers”, indicando la cantidad sobrante y el artículo involucrado.

Se desarrollaron también funciones que devuelven un diccionario con la información actualizada de un pedido o artículo, permitiendo que el front-end simplemente visualice los valores, optimizando la carga en los dispositivos.

Conexión Front-end y Back-end

A diferencia de la fase anterior, en esta vista no es factible usar *AJAX* para la conexión, ya que cualquier modificación debería reflejarse en tiempo real en todos los dispositivos conectados. Una solución basada en constantes llamadas al back-end sería ineficiente y no escalable.

Para superar este desafío, se adoptaron los *WebSockets* de *Django*, denominados *Channels*⁸. A través de un patrón de *Type-based publish/subscribe (TPS)*, se establece una conexión persistente entre el front-end y el back-end. Esta conexión se actualiza sólo cuando el back-end envía información, dando como resultado una solución más eficiente y adaptable.

3.5. Diseño e implementación de vistas para las colas

Tras la implementación de las vistas de TPV y la de pedidos pendientes, el siguiente paso es diseñar e implementar una vista para las colas de cada estación de trabajo. Esta vista se estructura en tres partes, similar a las dos vistas previamente mencionadas.

Front-end

Inicialmente, se planteó una idea simple: presentar columnas con listas de artículos, donde cada columna representa una estación de trabajo en la cocina. Además, se pensó en una opción que permitiera configurar fácilmente las estaciones a mostrar en el dispositivo, mediante casillas de verificación. Así, se desarrolló el siguiente mock-up:

⁸<https://websockets.readthedocs.io/en/stable/howto/django.html> (18 de junio de 2023)

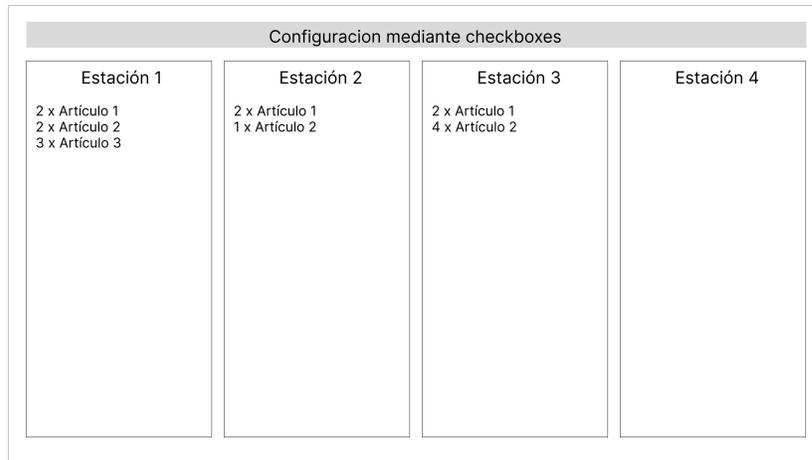


Figura 3.16: Mock-up de *display* de colas realizado en Figma.

Con el primer mock-up en mano, surgió un desafío: no había manera de identificar a qué pedido pertenecía cada artículo, ya que un artículo podría provenir de diversos pedidos. Adicionalmente, considerando que cada artículo tiene un estado, y con el objetivo de mantener la coherencia visual, se decidió usar los mismos colores que en las vistas 3.12, 3.13, 3.14, y 3.15 para indicar el estado de cada ítem.

Tras varias iteraciones sobre el diseño original, se llegó al siguiente mock-up:



Figura 3.17: Mock-up de *display* de colas plegables realizado en Figma.

No obstante, al intentar implementar este diseño, surgió un nuevo problema. Si se deseaban mostrar más de cinco estaciones, el tamaño de la fuente resultaba demasiado pequeño para ser legible en las tabletas de la cocina. Al aumentar el tamaño de la fuente, las palabras se deformaban. Una solución temporal fue limitar el número de estaciones seleccionables a

través de las casillas de verificación. Finalmente, se optó por presentar un máximo de ocho estaciones: cuatro en la parte superior y cuatro en la inferior. Aunque es poco probable tener solo una estación de armado de pedidos con tantas estaciones de preparación, si se necesitaran más estaciones, bastaría con añadir otra tableta para mostrarlas.

El diseño del prototipo final quedó de la siguiente manera:



Figura 3.18: Vista prototipo final de colas sin seleccionar ninguna estación de la aplicación.

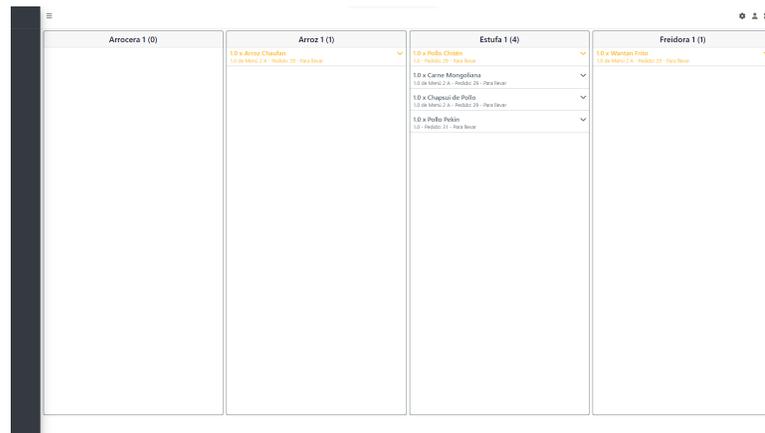


Figura 3.19: Vista prototipo final de colas seleccionando 4 estaciones de la aplicación.

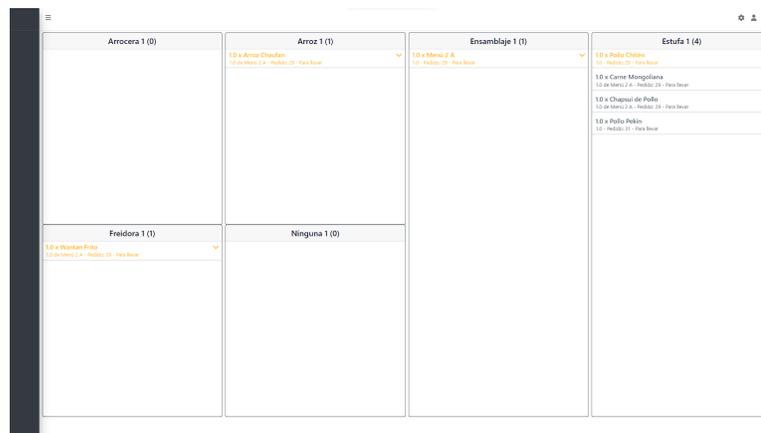


Figura 3.20: Vista prototipo final de colas seleccionando 6 estaciones de la aplicación.

Un detalle añadido en esta fase fue la capacidad de ocultar la barra de configuración con casillas de verificación para optimizar el espacio. Esta barra puede mostrarse nuevamente al presionar el ícono de engranajes en la esquina superior derecha. Además, se añadió el número de artículos cerrados junto al nombre de cada estación.



(a) Vista con barra de configuración.

(b) Vista sin barra de configuración.

Figura 3.21: Vista prototipo final de colas desde Galaxy Tab A 8” 2019 SM-T290 (1280 x 800 píxeles) de la aplicación.

Es importante destacar que la vista sin barra de configuración resulta más limpia y estéticamente agradable. Gracias a la funcionalidad incorporada, el usuario solo tendría que configurar la vista la primera vez, sin necesidad de ajustes posteriores a menos que se recargue la vista.

Back-end

De forma similar al back-end utilizado para la vista descrita en la sección 3.4, se desarrollaron funciones para recuperar y adaptar los datos requeridos por cada estación y artículo individualmente, permitiendo así invocar solo los objetos que se actualizan.

No obstante, durante este proceso se enfrentó un desafío significativo al tratar de implementar la función responsable de gestionar las colas. Inicialmente, se siguió el prototipo del

modelo de solución descrito en la tabla 3.9. Sin embargo, surgieron varios problemas con este prototipo: no se tuvieron en cuenta los pedidos programados, se omitió considerar el estado de los artículos, el cálculo de prioridad resultó ser más complejo de lo anticipado y los excedentes no se comportaban según lo esperado.

Para solucionar estos problemas, se ajustó el prototipo. En cuanto a los pedidos programados, se decidió solamente encolar aquellos que no estuvieran programados o que su programación no estuviera demasiado alejada del tiempo presente. A raíz de esto, fue necesario definir un tiempo estimado basado en una función modelada con diversas funciones dentro de la herramienta *GeoGebra*⁹. La función seleccionada no resultó excesivamente compleja y emulaba de forma bastante fiel la realidad: si uno cocina una cantidad x_1 en un tiempo y_1 , cocinar $x_2 = 2 * x_1$, no se debería de tomar $y_2 = 2 * y_1$ (en la mayoría de los casos). Esto indica que el tiempo no aumenta linealmente, sino que es algo menor.

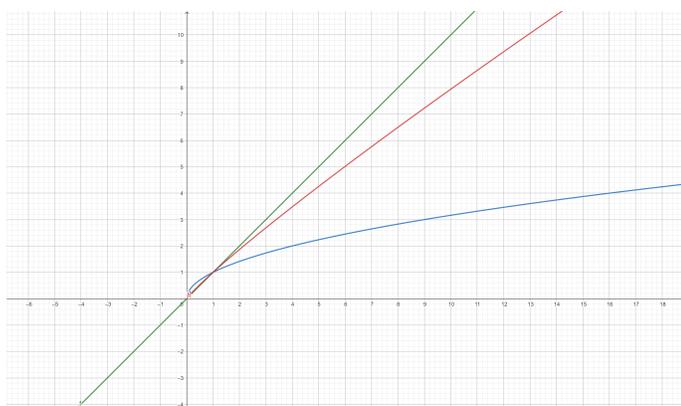


Figura 3.22: Funciones del tipo $f(x) = A * x^b$ vs. identidad graficada en *GeoGebra*.

En la figura, la función identidad se representa en verde, la función $f(x) = 1 * x^{0.9}$ en rojo y la función $g(x) = 1 * x^{0.5}$ en azul. Estas pueden interpretarse como si $f(x)$ y $g(x)$ fueran las estimaciones de tiempo para dos artículos diferentes, a_1 y a_2 . Si se tarda una unidad de tiempo en preparar una porción de cada uno, al preparar dos porciones, a_1 tomará algo más de tiempo que a_2 , pero ambos serán más rápidos que preparar los platillos unidad por unidad.

Utilizando una función como $f(x) = A * x^b$, es posible tener una función específica para cada platillo, ya que solo depende de dos parámetros: A que representa el tiempo de preparación de una porción, y b , un exponente definido.

Para determinar valores aproximados de A y b , se pueden usar muestras de tiempos y cantidades reales y aplicar una regresión lineal [1]. Sin embargo, para aplicar la regresión a la función 3.5, es necesario transformarla aplicando logaritmos en ambos lados:

$$\log(y) = \log(a) + b * \log(x) \tag{3.1}$$

Donde la ecuación 3.1 al ser de la forma $y = m + a * x$, se le puede realizar una regresión,

⁹<https://www.geogebra.org/?lang=en> (22 de junio de 2023)

sólo que en vez de ocupar los datos, se tendrían que sacar los logaritmos de los mismos antes de realizar la regresión y en vez de tener el valor exacto de a , tendríamos el valor de $\log(a)$, donde el logaritmo tendría la misma base que los logaritmos de la ecuación 3.1, por lo que simplemente faltaría hacer un cálculo y tendríamos el valor de a .

En cuanto a la gestión de los estados de los artículos, antes de incrementar la cantidad de un artículo en la cola, es esencial verificar que no esté en preparación. Para ello, se propuso agregar un campo en la base de datos que se actualice cuando el artículo esté al principio de la cola de una estación.

El modelo de datos original para determinar prioridades tenía múltiples niveles, complicando su gestión. Se optó por simplificarlo, creando una tabla con los campos “MenuFamily”, “OrderType” y “Priority”. Así, al agregar una instancia de una familia y tipo de pedido, esta se etiquetaría con su prioridad correspondiente. Cuanto mayor sea el valor en “Priority”, más alta será su prioridad en la cola. Debido a esta modificación, la estructura dejó de ser una cola de prioridad para convertirse en una lista ordenada. Así, un artículo con mayor prioridad estará más adelante, a menos que el primer artículo ya esté en preparación.

Con respecto a la propuesta de artículos más demandados mencionada en la sección 3.1.1, se decidió no implementarla directamente. En cambio, esta propuesta se puede reflejar ajustando el valor predeterminado mostrado en la figura 3.13. Si se desea que el programa sugiera cocinar excedentes, la visualización de las colas de estaciones tendría que mostrar la cantidad de excedente. Esto sobrecargaría una vista ya de por sí saturada de información, convirtiéndola en una potencial fuente de errores futuros.

Finalmente, se abordó un problema con excedentes de alimentos: si un artículo ya existía en la cola y surgían sobras, estas se atribuían incorrectamente, atribuyéndolos al pedido recién se está encolando, en vez de al primero de la fila. Para resolverlo, se propone asignar las sobras al momento de detectar un exceso, y sólo se crearía una instancia “LeftOvers” cuando se haya recorrido todas las colas, no haya más artículos del mismo tipo y aún quede excedente preparado.

El modelo de solución del prototipo final quedó de la siguiente manera:

Tabla 3.10: Modelo de solución del prototipo final.

Reglas
1. Encontrar el primer pedido p de la lista de pedidos lp que no ha sido encolado, pasar a 2. Si no existe p , terminar.
2. Encontrar el primer elemento e del pedido p que aún no esté encolado, si existe e , pasar a 3. Si no, marcar p como encolado y devolverse a 1.
3. Crear dos variables $t = e.tipo_articulo$ y $c = e.cantidad$. Pasar a 4.
4. Encontrar la lista de estaciones de trabajo let que sea del tipo de estación de trabajo de e . Pasar a 5.
5. Crear tres variables $len = infinito$, $let_i = 0$ y $ef = let[let_i]$. Pasar a 6.
6. Verificar si el largo de la cola de $let[let_i]$ es menor que len , si es menor, asignar ef como $let[let_i]$ len como el largo de ef . Pasar a 7.
7. Crear la variable $col_i = 0$ y $col = let[let_i].cola$. Pasar a 8.
8. Verificar si el estado $col[col_i]$ se esté preparando, si es así, pasar a 13, si no, pasar a 9.
9. Verificar si la prioridad de e es mayor que la de $col[col_i]$, si es mayor, se inserta un elemento a la cola col en la posición col_i con la prioridad de e y tipo de artículo t . Pasar a 10.
10. Verificar si la capacidad cap de la estación $let[let_i]$ dividido por la cantidad utilizada un de artículo de tipo t es mayor que la cantidad col_cant de $col[col_i]$, que la prioridad de e sea la misma prioridad que la de $col[col_i]$ y que el tipo de artículo de $col[col_i]$ sea t , si se cumplen las tres condiciones anteriores, pasar a 11, si no, pasar a 13.
11. Verificar si $c + col_cant > cap$, si es así, crear un subelemento de la cola con campos de cola = $col[col_i]$, artículo = e y cantidad = $cap - col_cant$ y redefinir $c = c + col_cant - cap$, pasar a 13, si no, pasar a 12.
12. Crear un subelemento de la cola con campos de cola = $col[col_i]$, artículo = e y cantidad = c , marcar e como encolado. Devolverse a 2.
13. Aumentar col_i en 1, si col_i es menor que el largo de la cola col , devolverse a 8, si no, pasar a 14.
14. Aumentar let_i en 1, si let_i es menor que el largo de la lista let , devolverse a 6, si no, pasar a 15.
15. Insertar un elemento $last$ al final de la cola de la estación ef con tipo de artículo t y prioridad de e , crear un subelemento con campos de cola = $last$, artículo = e y cantidad = $minimo(c, cap)$, redefinir $c = c - minimo(c, cap)$. Pasar a 16.
16. Verificar si $c = 0$, si es así marcar e como encolado, devolverse a 2, si no, modificar la cantidad de e por el nuevo valor de c y devolverse a 3.

Cada cola de estaciones representa una posición dentro de una instancia en la base de datos. Estas colas contienen sub-instancias de tipo cola, las cuales pueden representar una línea de pedido total o parcialmente.

Durante el proceso de desarrollo, se exploraron distintas estructuras preexistentes, incluidos los heaps. Estos se evaluaron en dos contextos diferentes. Al intentar implementar los heaps en las sesiones de *Django*, se evidenció que las colas no mantenían consistencia si los dispositivos iniciaban sesión con cuentas diferentes. Posteriormente, se consideró la creación de heaps a nivel global. Aunque la implementación de variables globales en desarrollo web no es recomendable, según un hilo del foro oficial de *Django*¹⁰, se decidió probar esta estrategia.

¹⁰<https://forum.djangoproject.com/t/global-variable-not-updating-in-django-view/1941> (17 de julio de

Aunque en principio funcionó correctamente, la vulnerabilidad ante interrupciones, como un corte de electricidad, podría causar la pérdida de las colas, generando inconsistencias.

Basándonos en las recomendaciones del foro y en las limitaciones observadas, especialmente porque las variables en las vistas podrían no actualizarse adecuadamente, se decidió almacenar las colas en la base de datos, tratándolas como listas simples. Dado que estas colas se guardan y recuperan de una base de datos, la diferencia de rendimiento entre acceder a un heap y a una lista se torna casi insignificante, ya que el acceso o escritura en la base de datos se convierte en el paso más tardado.

Un factor adicional a considerar es que, para insertar un elemento en las colas, es esencial revisar todos los elementos de las colas existentes para verificar su estado. Por lo tanto, no se aprovecharía el principal beneficio de un heap, que es el acceso en $O(1)$ al elemento de mayor o menor valor. Esto hizo que la implementación de un heap resultara innecesaria.

Conexión Front-end y Back-end

Para establecer la conexión entre el front-end y back-end en esta etapa, se empleó el procedimiento descrito en la sección 3.4. Se utilizó el sistema de *WebSockets* ya implementado. Sin embargo, se adaptó la función responsable de enviar paquetes para la visualización de pedidos, permitiendo que ahora enviara un paquete adicional con la información pertinente de las estaciones que hubieran experimentado algún cambio.

3.6. Modelo de datos del prototipo final

A lo largo del proceso de desarrollo, la base de datos experimentó diversas modificaciones. Los ajustes más notables incluyeron la incorporación de las tablas “WorkStationQueue” y “QueueElement” para gestionar las colas de cada estación, y la tabla “Priorities” que detalla las prioridades de los artículos en el menú. También se introdujeron las tablas “SideNote” y “OrderItemNote” para manejar las anotaciones asociadas a los artículos, y se ajustaron algunas tablas preexistentes. Optamos por reducir las tablas de estado a un simple campo, dado que no resultaba práctico tener tablas con un único campo. El modelo resultante se ilustra a continuación:

2023)

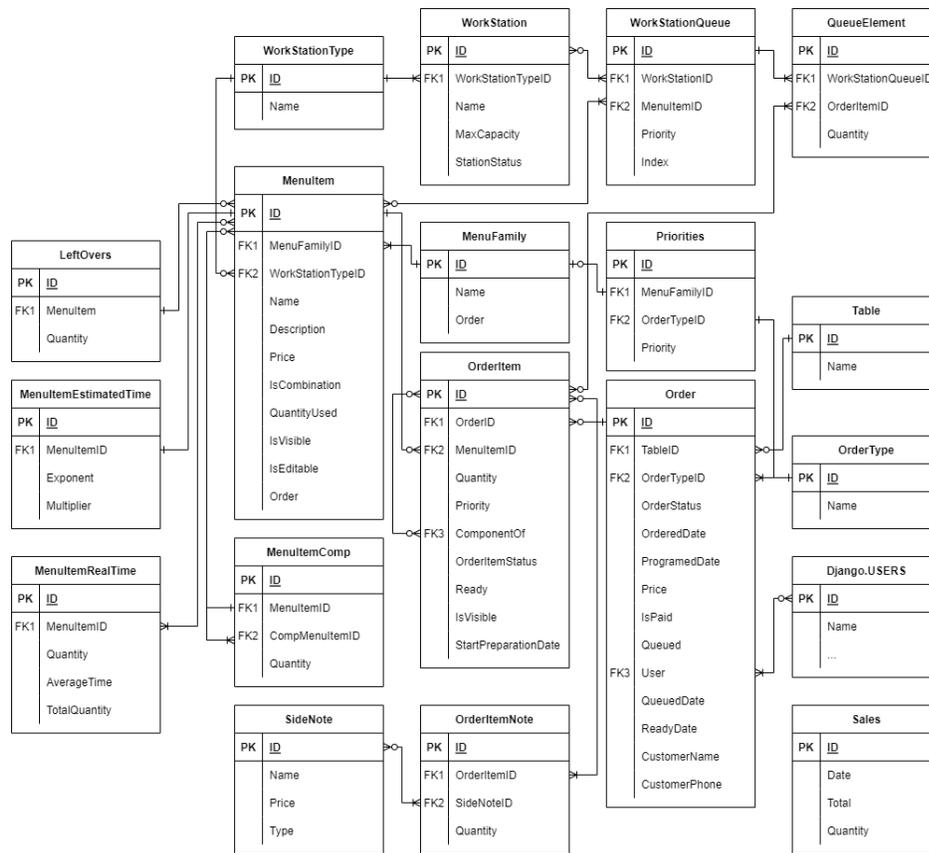


Figura 3.23: Modelo de datos Final.

Estas modificaciones surgieron a medida que la aplicación evolucionaba. Por ejemplo, los campos “Priority” en las tablas “MenuFamily”, “MenuItem” y “OrderType” se adaptaron debido a cambios en el algoritmo de cálculo de prioridad, que se detallará más adelante.

La necesidad de las tablas “SideNote” y “OrderItemNote” emergió al implementar un sistema TPV, ya que comúnmente estos programas permiten agregar anotaciones con valores adicionales, ya sea de tipo booleano o una cantidad específica. La tabla “SideNote” representa cada tipo de nota que se tenga disponible en para los artículos del menú, mientras que la tabla “OrderItemNote” representa las notas de un elemento de un pedido.

Se enriquecieron las tablas “MenuItem”, “OrderItem” y “Order” para una mejor especificación de cada elemento. Particularmente, los campos “QuantityUsed” de “MenuItem” que viene siendo el espacio utilizado por el elemento dentro de la estación y “ComponentOf” de “OrderItem” es un campo para representar la composición de elementos dentro de otro elemento, cantidad preparada “Ready”, “Quantity”, “Priority”, entre otros campos.

Otro cambio fueron la creación de campos de “ReadyDate” y “QueuedDate” de la tabla “Order” para tener un registro de cuando un pedido era encolado hasta cuando estaba listo para la obtención de métricas, el campo “StartPreparationDate” de la tabla “OrderItem” para tener registro de cuando un elemento empieza a prepararse. Y la tabla “Sales” para tener registro de las ventas del día, funcionalidad solicitada durante el desarrollo por algunos de expertos entrevistados anteriormente.

Adicionalmente, se incorporaron campos como “ReadyDate”, “QueuedDate” y “StartPreparationDate” para tener el seguimiento de pedidos y elementos y generar métricas útiles. La tabla “Sales” para registrar las ventas diarias, una funcionalidad sugerida por expertos consultados.

Finalmente, las tablas “MenuEstimatedTime” y “MenuItemRealTime” se relacionan con el tiempo de preparación de los artículos, representando la ecuación 3.5. Se proponen ecuaciones para optimizar el cálculo del tiempo medio, donde el campo “Multiplier” corresponde a la variable a y “Exponent” a la variable b . Mientras que la tabla “MenuItemRealTime” corresponde a un registro de los valores reales del tiempo que se demora en preparar una cantidad “Quantity”, realizando unos cálculos para no tener que guardar todas las instancias individualmente, y en vez guardarlos por la cantidad producida y un promedio de los tiempos, llegando a las siguientes definiciones:

$$AverageTime = \frac{AverageTime * TotalQuantity + \Delta Time}{TotalQuantity + 1} \quad (3.2)$$

$$TotalQuantity = TotalQuantity + 1 \quad (3.3)$$

Donde de la ecuación 3.2, el primer factor del numerador ($AverageTime * TotalQuantity$) es el tiempo medio por el número total de registros que se tendrían si se registrase en una tabla, mientras que $\Delta Time$ es el tiempo que se quisiera registrar y el denominador sería el nuevo $TotalQuantity$.

Sin embargo, con la ecuación 3.2 el factor $AverageTime * TotalQuantity$ puede llegar a ser muy grande cuando $TotalQuantity$ sea demasiado grande, provocando un *overflow*, para evitar un posible error, se realizaron las siguientes operaciones:

$$AvegareTime = \frac{AverageTime * (TotalQuantity + 1 - 1) + \Delta Time}{TotalQuantity + 1}$$

$$AvegareTime = \frac{AverageTime * (TotalQuantity + 1) - AverageTime + \Delta Time}{TotalQuantity + 1}$$

$$AvegareTime = AverageTime + \frac{\Delta Time - AverageTime}{TotalQuantity + 1} \quad (3.4)$$

3.7. Estructura general de la aplicación

La aplicación inicia su flujo de operación con la introducción de un pedido a través de la interfaz TPV. Una vez se registra el pedido en la base de datos, se clasifica: si es un pedido programado, queda en estado pendiente; de lo contrario, todos sus elementos se encolan en sus respectivas estaciones.

Luego, en la vista de pedidos pendientes (3.4), el nuevo pedido se sitúa al final de la lista y sus artículos aparecen en la interfaz de colas, siempre que se hayan encolado previamente.

Cuando los artículos están listos, los usuarios pueden marcarlos como “Listo” en la vista de pedidos pendientes y registrar la cantidad preparada. Si la cantidad introducida excede la requerida, el sistema verifica si hay otros pedidos que necesiten el mismo artículo, ajustando la cantidad en consecuencia. Esto puede ocasionar dos escenarios: si la cantidad final preparada es cero, no se realiza ninguna acción; pero si es positiva, se interpreta como un excedente y se genera o incrementa una entrada de sobras para ese artículo. Durante este proceso, se actualizan los estados de los artículos afectados, indicando si están listos o no.

Una vez procesados los cambios, se recuperan los identificadores de los artículos modificados y se transmiten a una función encargada de generar las actualizaciones necesarias desde el back-end. Estas actualizaciones se envían a través de *WebSockets* a todas las vistas conectadas.

Finalmente, al recibir los datos desde el *WebSocket*, las vistas ejecutan las actualizaciones correspondientes para reflejar la nueva información.

Capítulo 4

Evaluación de la solución

En este capítulo se definen los procedimientos para poder evaluar la solución construida.

4.1. Definición de la evaluación

Como la aplicación la forman tres interfaces, dos de las tres funcionan juntos para mostrar la información necesaria en la cocina, y la tercera se centra en el ingreso de los pedidos. Se probaron ambos conjuntos por separado. De esta manera, se definen dos perfiles de usuarios: el de tipo “cajero/cajera”, que se encarga de ingresar los pedidos desde la interfaz de toma de pedidos desarrollada, y el de tipo “ensamblador/ensambladora”, que se encarga de armar los pedidos e indicar el orden de preparación de los artículos interactuando con las interfaces de las colas de las estaciones y la de los pedidos pendientes. Este último perfil corresponde a la persona o personas con mayor interacción con los pedidos dentro de una cocina.

Adicionalmente, dado que el plazo total para la realización de este trabajo de título es únicamente de 15 semanas, se decidió emplear de 3 a 4 semanas para realizar todas las pruebas necesarias para las pruebas requeridas para la recopilación de datos.

4.2. Protocolo de experimentación

En esta sección se presentan los protocolos que se tomaron en cuenta al realizar las pruebas, tanto para la vista de toma de pedidos desarrollada como para las vistas de *display* de información de los pedidos.

4.2.1. Protocolo para vista TPV

Para la evaluación de la interfaz de toma de pedidos, se diseñó un protocolo que consiste en una prueba basada en la realización de tareas dentro de la aplicación por parte del usuario. En estas, se midió una variante de las métricas de usabilidad de Wilson y Wixon [4], con el fin de obtener una medida cuantitativa de la prueba. Tras la experiencia, el usuario debió responder a una encuesta para medir la complejidad percibida y posibles mejoras en el diseño de la vista.

Participantes

¿Quiénes conformaron la muestra?

La muestra estuvo conformada por usuarios con diferentes niveles de experiencia en la toma de pedidos en varios restaurantes de comida china, tanto para llevar como para servir en el lugar. Esta muestra incluyó a cajeros, asistentes de cocina, cocineros y camareros. Los usuarios debían tener al menos un conocimiento básico de la estructura del menú de un restaurante de comida china, ya que los datos utilizados durante la prueba eran de un menú de este tipo.

¿Cómo se reclutaron los participantes?

Dado que el sistema desarrollado no está centrado en la toma de pedidos, sino que solo es un medio para ingresarlos, no se hizo tanto hincapié en la prueba de esta fase, por lo que se reclutó un número reducido de participantes. Invitamos a personas conocidas del estudiante de maestría que trabajaran en la industria de la comida china, tanto trabajadores de varios restaurantes como amigos y familiares del estudiante.

Criterios de inclusión y exclusión

El criterio de inclusión fue que el usuario tuviera experiencia previa trabajando en restaurantes de comida china y tuviera conocimiento o noción de la estructura del menú de un restaurante de este tipo. El criterio de exclusión se aplicó a cualquier otro usuario que no cumpliera con el perfil anterior.

Categorización

- Género: Cualquiera.
- Edad: Cualquiera.
- Conocimiento previo de la aplicación: Nulo o bajo.
- Conocimiento previo a programas TPV: Cualquiera.
- Conocimiento previo necesario: Como mínimo tener una idea de la estructura de una carta de un restaurante de comida china.

Verificabilidad de la muestra

Dado que los participantes son trabajadores de distintos restaurantes de comida china, se espera que, al menos, tengan un conocimiento general de la estructura de un menú de un restaurante de comida china. Por ejemplo, deberían saber que el artículo “Arrollado Primavera” puede encontrarse en la sección de “Frituras” o “Aperitivos”, y que el artículo “Carne Mongoliana” puede encontrarse en la sección de “Vacuno” o “Carnes”.

Consideraciones éticas

No presionar a los usuarios de una forma que los pueda incomodar.

Posibles sesgos en la muestra

Dado que la muestra es relativamente pequeña y no requiere un amplio conocimiento previo en este tipo de aplicaciones, solo tener una noción de la estructura básica de la carta de un restaurante de comida china, y se reclutaron a los usuarios por trabajar en restaurantes chinos, el único sesgo que podría tener la muestra es que los usuarios no estén familiarizados con la carta del restaurante en el que trabajan o han trabajado. Para ello, las pruebas se diseñaron con platos populares, esperando que los participantes puedan completar sin mayores complicaciones.

Tamaño de la muestra

Debido a que el sistema a probar no es el tema central del proyecto de maestría, dedicamos solo tres días para reclutar y encuestar a la muestra total. El tamaño final de la muestra para este experimento fue de 10 personas.

Definición del experimento

La prueba consistió en la realización de tareas definidas por un sujeto externo sin capacitación previa, con el fin de estudiar la usabilidad e intuitividad de la interfaz.

Proceso de experimentación

Para el reclutamiento de participantes, se invitó a personas conocidas del memorista que cumplieran con el perfil requerido.

Seguido de entregarle una breve explicación acerca del proceso en grandes rasgos, donde un parafraseo de la explicación fue: “Es una aplicación para la toma de pedidos, en la que se puede modificar menús, agregar notas de manera directa, agregar información del cliente e información del pedido. Le pediré que ingrese unos pedidos dentro del sistema y que durante el proceso me vaya indicando lo que va sintiendo y viendo. No hay necesidad de sentir presión, no estamos evaluando a usted, sino que a la aplicación.”.

Luego de eso, se dio paso a la realización de una serie de tareas desde un dispositivo con la interfaz cargada de antemano, tareas que corresponden al ingreso de diversos pedidos, donde las tareas fueron:

1. Primera tarea:

Acciones a ejecutar:

- Ingresar “1 x Menú para 2A”.
- Presionar “Total” gris.
- Presionar “Total” verde.

2. Segunda tarea:

Acciones a ejecutar:

- Ingresar “1 x Menú para 2A”.
- Ingresar “1 x Menú para 2B”.
- Eliminar “1 x Menú para 2A”.
- Presionar “Total” gris.
- Presionar “Total” verde.

3. Tercera tarea:

Acciones a ejecutar:

- Ingresar “1 x Carne Mongoliana”.
- Presionar “Total” gris.
- Ingresar “Juan” como el nombre de cliente y “12345678” como el teléfono del cliente.
- Presionar “Total” verde.

4. Cuarta tarea:

Acciones a ejecutar:

- Ingresar “1 x Menú para 2A”.
- Agregar nota “Sin Sal” a “Menú para 2A”.
- Presionar “Total” gris.
- Presionar “Total” verde.

5. Quinta tarea:

Acciones a ejecutar:

- Ingresar “1 x Menú para 2A”.
- Eliminar “1 x Carne Mongoliana” de “Menú para 2A”.
- Ingresar “Pollo Mongoliano”.
- Presionar “Total” gris.
- Presionar “Total” verde.

6. Sexta tarea:

Acciones a ejecutar:

- Ingresar “1 x Menú para 2A”.
- Cambiar tipo de pedido por “Mesa”.
- Seleccionar “Mesa 3”.
- Presionar “Guardar” verde.
- Presionar “Nuevo pedido” en la barra de navegación.

7. Séptima tarea:

Acciones a ejecutar:

- Ingresar “Chapsui de Pollo”.
- Presionar “Total” gris.
- Presionar “Total” verde.

Después de completar todas las pruebas, se administró una breve encuesta, la cual incluyó preguntas sobre la complejidad percibida y posibles mejoras en el diseño de la interfaz.

Proceso repetido a lo largo de tres días, llegando a un total de diez participantes.

Instrumentos de Recolección de Datos

Tipo de estudio

Caso de estudio exploratorio, debido a que prioriza los puntos de vista de las personas. La encuesta post-intervención fue realizada mediante una encuesta en *Google Forms*. Con

respecto a la medición de las métricas, el memorista estuvo atento al proceso, para ir contando errores y ayudas.

¿Cómo se recolectó la información?

Se hizo recolección manual de los datos, completando un formulario en *Google Forms*, donde iba almacenando los datos de manera digital en una hoja de cálculo de *Google Sheets*, para hacer los estudios necesarios.

Calibración de los instrumentos

Como los instrumentos son meramente encuestas en un formulario en línea, no requieren de una mayor calibración, únicamente hubo que fijarse que el dispositivo con el formulario estuviera cargado.

Validez y confianza de los instrumentos

El método de Wilson y Wixon [4] es un método bien conocido, estudiado y validado. Por otro lado, la segunda encuesta sirve para evaluar cómo los participantes perciben la aplicación y para obtener retroalimentación de los mismos.

Limitaciones de los instrumentos: validez interna y externa

Por el único motivo de que los dispositivos que ejecutaban la aplicación eran dispositivos que poseía el memorista desde antes de empezar el desarrollo, no se probó con otro dispositivo para la ejecución de la aplicación.

Procedimiento de Recolección de Datos

¿Cómo se recolectó la información?

En ambas instancias (durante y post-intervención) se registraron mediante un formulario creado de antemano en *Google Forms* que registran las respuestas en una hoja de cálculo en *Google Sheets*, sólo que el formulario para recolectar datos durante la intervención fue respondido por el memorista, mientras que el formulario post-intervención fue respondido por el usuario tras haber realizado las tareas.

Condiciones de observación y/o experimentación

El usuario y el memorista debían estar en el mismo lugar durante la prueba. Al momento de comenzar a utilizar la aplicación, el usuario debía estar un poco alejado del memorista, a menos que necesitara ayuda.

Pilotaje del proceso

No se realizó ningún tipo de pilotaje debido a que el largo de prueba en sí es relativamente corta, sumado a que la cantidad de usuarios reclutados tampoco es muy grande, se encontró innecesario realizar un pilotaje para este proceso.

Reproducibilidad del proceso

El proceso es reproducible, ya que se especifican las componentes y los instrumentos utilizados en el anexo.

4.2.2. Protocolo para vistas de información de pedidos

Para la evaluación de las interfaces de *display* de información de pedidos, se diseñó un protocolo que consiste en dos fases. Una primera basada en la realización de tareas dentro de la aplicación bajo un espacio controlado, por parte del memorista (fase A). Y una segunda fase basada en la instalación del sistema dentro de un restaurante de comida china durante una semana (fase B) e ir registrando los errores tanto del orden de las colas como el resto de los errores que pudieran surgir, donde el memorista se encargará de ingresar los pedidos desde la caja, mientras que los trabajadores se encargaran de interactuar con dos dispositivos dentro de la cocina con las interfaces de pedidos pendientes y colas de estaciones cargadas de antemano.

Definición del experimento fase A

Prueba de casos simulados, que consistían en la ejecución de una lista de acciones en un orden específico junto a un resultado esperado o un conjunto de resultados posibles. Optando por realizar múltiples veces cada prueba, esto debido a que podría ocurrir un error inesperado por parte del sistema durante la ejecución algunas pruebas, así al realizar una prueba varias veces, se tendría mayor certeza de si el sistema tiene el comportamiento esperado o no.

Valores iniciales

Debido a que el algoritmo al que se quiere verificar su comportamiento esperado depende no sólo de los inputs, sino que también de los valores de variables limitantes, como por ejemplo la cantidad máxima de alimentos que se puede preparar al mismo tiempo dentro de una estación, siendo un valor posiblemente diferente para cada estación, las pruebas se diseñaron teniendo en cuenta la siguiente configuración inicial:

Tabla 4.1: Estaciones de trabajo para pruebas.

Estación	Tipo	Capacidad Máxima
Platillos_01	1	6
Platillos_02	1	6
Frituras_01	2	6
Arroz_01	3	6

Tabla 4.2: Artículos para pruebas.

Artículos	Familia	Tipo Estación	Cantidad Utilizada
Carne Mongoliana	Vacuno	1	1
Chapsui de Pollo	Verduras	1	1
Wantan	Aperitivos	2	2
Arrollado Primavera	Aperitivos	2	1
Arroz Chaufan	Arroz	3	1

Tabla 4.3: Prioridades para pruebas.

Familia	Tipo de Pedido	Prioridad
Aperitivos	Servir	1

Tabla 4.4: Combinaciones para pruebas.

Artículo	Sub Artículo	Cantidad
Menú para 2A	Wantan	1
Menú para 2A	Carne Mongoliana	1
Menú para 2A	Chapsui de Pollo	1
Menú para 2A	Arroz Chaufan	2

Donde la estación Platillos_02 sólo se activó para algunas pruebas en donde se quería probar los casos en que hubiese más de una estación del mismo tipo.

Proceso de experimentación

Se comienza con el ingreso de una serie de pedidos desde la interfaz TPV desarrollada, para luego verificar que coincida con el resultado esperado dependiendo del caso en específico, siendo pruebas binarias, es decir, se pasaban o no se pasaban. Donde algunas de las pruebas son:

Prueba 1, pedidos por orden de llegada:

Tabla 4.5: Prueba 1 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar
2	2	Para llevar

Tabla 4.6: Prueba 1 (artículos).

Cantidad	Artículo	Pedido
1	Carne Mongoliana	1
3	Carne Mongoliana	2

Tabla 4.7: Prueba 1 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Platillos_1	1	Preparando	Carne Mongoliana	<ul style="list-style-type: none"> • 1 x Pedido: 1.
Platillos_1	3	Encolado	Carne Mongoliana	<ul style="list-style-type: none"> • 3 x Pedido: 1.

Donde el orden esperado es el de 4.7, pues al llegar el primer pedido, se debería de encolar el primer artículo de inmediato, luego cuando llega el segundo pedido, al tratar de encolarlo, como el primer elemento de la cola está en preparación, se agregará un nuevo elemento a la cola de la estación.

Prueba 2, superando el límite de una estación:

Tabla 4.8: Prueba 2 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar

Tabla 4.9: Prueba 2 (artículos).

Cantidad	Artículo	Pedido
15	Carne Mongoliana	1

Tabla 4.10: Prueba 2 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Platillos_1	6	Preparando	Carne Mongoliana	• 6 x Pedido: 1.
Platillos_1	6	Encolado	Carne Mongoliana	• 6 x Pedido: 1.
Platillos_1	3	Encolado	Carne Mongoliana	• 3 x Pedido: 1.

Donde el orden esperado es el de 4.10, pues se añade un elemento que supera el límite que se puede cocinar en la estación correspondiente, que en este caso es de 6, se espera que se encolen 3 veces, 2 veces con el máximo y una vez con 3.

Prueba 3, encolar aperitivos de pedidos para servir:

Tabla 4.11: Prueba 3 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar
2	2	Para servir

Tabla 4.12: Prueba 3 (artículos).

Cantidad	Artículo	Pedido
1	Wantan	1
1	Arrollado Primavera	1
1	Wantan	2

Tabla 4.13: Prueba 3 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Freidora_1	1	Preparando	Wantan / Arrollado Primavera	• 1 x Pedido: 1.
Freidora_1	1	Encolado	Wantan	• 1 x Pedido: 2.
Freidora_1	1	Encolado	Arrollado Primavera / Wantan	• 1 x Pedido: 1.

Donde el orden esperado es el de 4.13, pues como se ingresa un pedido para llevar con una porción de “Wantan” y una de “Arrollado Primavera” antes que un pedido para servir con una porción de “Wantan”, primero se encola una porción de “Wantan” o una de “Arrollado Primavera”, el orden en este caso es aleatorio, pues es indiferente en que orden se preparen en la vida real, basta con que se preparen ambos para poder entregar el pedido al cliente, luego se encola el artículo que no se encoló antes, por último, como se le da prioridad a los aperitivos para pedidos para servir, al tratar de encolar el segundo pedido, se debería encolarse lo antes posible, que en este caso sería justo después del elemento que se esté preparando.

Prueba 4, encolar combinaciones de manera correcta:

Tabla 4.14: Prueba 4 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar

Tabla 4.15: Prueba 4 (artículos).

Cantidad	Artículo	Pedido
1	Menú para 2A	1

Tabla 4.16: Prueba 4 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Freidora_1	1	Preparando	Wantan	• 1 x Pedido: 1.
Arroz_1	2	Preparando	Arroz Chaufan	• 2 x Pedido: 1.
Platillos_1	1	Preparando	Carne Mongoliana / Chapsui de Pollo	• 1 x Pedido: 1.
Platillos_1	1	Encolado	Chapsui de Pollo / Carne Mongoliana	• 1 x Pedido: 1.

Donde el orden esperado es el de 4.16, pues simplemente basta con separar los elementos que componen al elemento de la combinación y encolarlos en sus respectivas estaciones, adicionalmente, al igual que en la prueba anterior, es indiferente si la “Carne Mongoliana” o “Chapsui de Pollo” está encolado antes el uno que el otro.

Prueba 5, encolar con más de una estación del mismo tipo:

Tabla 4.17: Prueba 5 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar
2	2	Para llevar

Tabla 4.18: Prueba 5 (artículos).

Cantidad	Artículo	Pedido
1	Carne Mongoliana	1
1	Carne Mongoliana	2
1	Chapsui de Pollo	2

Tabla 4.19: Prueba 5 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Platillos_X	1	Preparando	Carne Mongoliana	• 1 x Pedido: 1.
Platillos_Y	1	Preparando	Carne Mongoliana / Chapsui de Pollo	• 1 x Pedido: 2.
Platillos_(X o Y)	1	Encolado	Chapsui de Pollo / Carne Mongoliana	• 1 x Pedido: 2.

Donde el orden esperado es el de 4.19, pues al ingresar el primer pedido, se trata de encolar los artículos a sus estaciones respectivas, pero como hay más de una estación a la que se pueden encolar, se tiene que escoger la cola más corta para encolar el artículo, que en un principio, como ambas estaciones están vacías, es indiferente en cuál estación se encolara primero, luego al llegar el segundo pedido, es indiferente qué artículo se encola primero, sin embargo, se tiene que encolar en la otra estación que está vacía, por último, se encola el artículo que no se encoló anteriormente en cualquiera de las dos estaciones.

Prueba 6, superando el límite de una estación 2:

Tabla 4.20: Prueba 6 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar

Tabla 4.21: Prueba 6 (artículos).

Cantidad	Artículo	Pedido
15	Wantan	1

Tabla 4.22: Prueba 6 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Frituras_1	3	Preparando	Wantan	• 3 x Pedido: 1.
Frituras_1	3	Encolado	Wantan	• 3 x Pedido: 1.
Frituras_1	3	Encolado	Wantan	• 3 x Pedido: 1.
Frituras_1	3	Encolado	Wantan	• 3 x Pedido: 1.
Frituras_1	3	Encolado	Wantan	• 3 x Pedido: 1.

A diferencia de la prueba 2, en este caso se encola un artículo que ocupa más espacio dentro de una estación, es decir, se puede preparar una menor cantidad del mismo elemento al mismo tiempo, como la cantidad de espacio utilizada por una porción de “Wantan” está definida como 2, entonces se deberían de encolar de 3 en 3 porciones, pues como el límite de la estación es de 6 y la cantidad de espacio que se necesita para una porción de “Wantan” es de 2, el máximo sería de 3 porciones.

Prueba 7, pedidos programados:

Fecha actual: 01-01-2023 13:00

Tabla 4.23: Prueba 7 (pedidos).

Pedido	Orden de llegada	Tipo	Programado
1	1	Para llevar	02-01-2023 13:00
2	2	Para llevar	-

Tabla 4.24: Prueba 7 (artículos).

Cantidad	Artículo	Pedido
1	Carne Mongoliana	1
1	Carne Mongoliana	2

Tabla 4.25: Prueba 7 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Platillos_1	1	Preparando	Carne Mongoliana	<ul style="list-style-type: none"> • 1 x Pedido: 2.

Donde sólo se espera que se encolen los artículos de los pedidos que están cerca de ser solicitados, en este caso, como el tiempo que falta para que se pase hasta el tiempo programado del pedido 1 es demasiado, no es necesario encolarlo de momento.

Prueba 8, colas con múltiples orígenes:

Tabla 4.26: Prueba 8 (pedidos).

Pedido	Orden de llegada	Tipo
1	1	Para llevar
2	2	Para llevar
3	3	Para llevar

Tabla 4.27: Prueba 8 (artículos).

Cantidad	Artículo	Pedido
1	Carne Mongoliana	1
3	Carne Mongoliana	2
4	Carne Mongoliana	3

Tabla 4.28: Prueba 8 (resultado esperado).

Estación	Cantidad	Estado	Tipo Artículo	Pedidos de origen
Platillos_1	1	Preparando	Carne Mongoliana	<ul style="list-style-type: none"> • 1 x Pedido: 2.
Platillos_1	6	Encolado	Carne Mongoliana	<ul style="list-style-type: none"> • 3 x Pedido: 2. • 3 x Pedido: 3.
Platillos_1	1	Encolado	Carne Mongoliana	<ul style="list-style-type: none"> • 1 x Pedido: 3.

Donde el orden esperado es el de 4.28, pues al ingresar el primer pedido se encola el artículo sin problemas, luego al encolar el segundo, como el elemento de la cola está en preparación, se tiene que dejar como encolado con la cantidad correspondiente, por último, al encolar el tercer pedido, como aún no se llega al límite de la estación, se debe dejar como cantidad a cocina como el límite de la estación, pues se supera si se suma a la cantidad actual, teniendo que crear un nuevo elemento en la cola con la cantidad restante.

El resto de las pruebas son combinaciones de las pruebas anteriores, por ejemplo, ingresar múltiples Menús para 2A a la vez teniendo múltiples estaciones. Todas las pruebas que se emplearon para la experimentación se encuentran en el anexo.

Instrumentos de Recolección de Datos experimento fase A

Tipo de estudio

Caso de estudio basándose en simulaciones en un ambiente controlado, mediante casos de prueba previamente definidos con un resultado esperado, donde un caso sólo puede ser aprobado o reprobado.

¿Cómo se recolectó la información?

Cada vez que se realizaba una prueba, se registraba en la misma hoja de cálculo en donde estaba registrada la prueba si se pasaba o no durante las sesiones de pruebas.

Calibración de los instrumentos

Al emplear únicamente un libro de cálculo de *Google Sheets* sin necesidad de medir ningún dato, no hubo necesidad de realizar ninguna calibración.

Limitaciones de los instrumentos: validez interna y externa

Al ser casos de pruebas creados de antemano, es complicado abarcar cada caso posible, por ello se pensaron casos que ocurren normalmente en el restaurante donde se probó el sistema, por ello, la única limitante del proceso es la creación de las pruebas como tal.

Procedimiento de Recolección de Datos fase A

¿Cómo se recolectó la información?

Al ser casos de pruebas, la información corresponde a si se pasa o no un caso de prueba, información recolectada según se iban realizando las pruebas.

Condiciones de observación y/o experimentación

El memorista se encontraba sólo realizando las pruebas una por una, registrando el resultado de cada prueba una vez realizada.

Pilotaje del proceso

Debido a que el proceso en sí consiste en seguir instrucciones y ver los resultados por parte del memorista, no se realizó ningún tipo de pilotaje.

Reproducibilidad del proceso

El proceso es reproducible, siempre y cuando se tengan los mismos artículos con familias, la tabla de prioridades, la tabla de estaciones, se podrían emplear las mismas pruebas, o bien, se pueden definir unas condiciones iniciales totalmente distintas, implicando tener que definir

nuevas pruebas. Las pruebas diseñadas se encuentran en el anexo.

Definición del experimento fase B

Prueba en casos reales, consiste en la instalación de la aplicación dentro de un restaurante de comida china para servir y para llevar, proceso que se llevó a cabo por más de una semana, registrando la cantidad de errores y los tipos de errores que surgían durante el período de esta fase.

Seguido de una fase de recopilación de datos de control en el mismo restaurante sin el sistema para realizar una comparación entre ambos casos.

Donde la variable principal para la comparación es el tiempo medio que toma preparar un pedido desde que es ingresado. Sin embargo, comparar los tiempos sin tener en cuenta el contexto de un pedido no sería correcto, para ello también se registrarán la cantidad de pedidos que estén en cola antes de un pedido, la cantidad de artículos y la cantidad de artículos diferentes de un pedido.

Por otro lado, solo estos datos no servirán para comprender el contexto de un pedido, pues hay artículos con mayor tiempo de preparación que otros. Si quisiéramos realizar un estudio mucho más detallado, se tendría que probar por semanas en ambos casos, para tener más datos y que los tiempos de preparación de cada platillo no sean determinantes en la comparación. Sin embargo, como El objetivo de este estudio es ver que el tiempo que toma preparar los artículos según el sistema es similar o menor que el tiempo que toma los artículos sin el sistema, no es necesario realizar un estudio que tomase tanto tiempo, sin considerar el limitado para el desarrollo del trabajo de título.

Proceso de experimentación

Considerando que se quiere probar las vistas dentro de la cocina, y que el memorista estaba ingresando los pedidos desde la caja del establecimiento, él mismo no podría estar en la cocina en todo momento, por ello, se capacitaron a dos personas que trabajaban en el restaurante donde se probó para ayudar a probar el sistema, además de que avisasen al memorista de los errores que surgían en cuanto se dieran cuenta.

Además, si el sistema deja de funcionar durante el día, se ingresaron los pedidos de igual manera en la aplicación de toma de pedidos que ya tenía el restaurante, que saca tickets de venta que se llevaron dentro de la cocina si el programa deja de funcionar. Sin embargo, se indicó que trataran de no emplear los tickets mientras la aplicación funcionara.

A medida que se presionaba los botones de “Listo” en la vista de pedidos, el sistema registra el tiempo total desde que el pedido fue realizado hasta el término del mismo.

Por otra parte, tras finalizar el día, el sistema tiene un sistema de cierre para guardar los tiempos medios que se demoraron los pedidos y la preparación de X cantidad de algún artículo, estadísticas que se contrastarán con datos en donde no esté el sistema instalado.

Terminada la prueba del sistema dentro del restaurante, se pasó una semana recopilando datos dentro del restaurante para tener datos de control con el que comparar los resultados

recopilados anteriormente.

Para ello, se aprovechó de que los tickets que emite el sistema dentro del restaurante, vienen con la fecha impresa a la hora de sacar los tickets, sin embargo, en el restaurante no se emiten los tickets de las mesas una vez ingresados, debido a que es posible que el pedido se modifique, para estos casos, como el memorista es el que ingresaba los pedidos, él ingresaba en una hoja de cálculo de *Google Sheets* la hora de ingreso del pedido y el número de la mesa; una vez que él se hubieran sacado todos los platillos del pedido, se ingresaba la hora de término.

Instrumentos de Recolección de Datos experimento fase B

Tipo de estudio

Caso de estudio basándose en una prueba en un ambiente real, mediante la instalación del sistema dentro de un restaurante de comida china real.

¿Cómo se recolectó la información?

Para registrar los errores de la aplicación, cada vez que ocurría un imprevisto por parte del sistema dentro del establecimiento, si era un nuevo error, se registraba una breve descripción del error en una tabla en *Google Sheets*, si era un error que ya había ocurrido, aumentaba el contador del error.

En cuanto al tiempo que transcurría desde el ingreso de un pedido hasta el término de la preparación del mismo, se realizaba de manera automática por parte del sistema, almacenando los datos dentro de la base de datos del sistema.

Por último, para la recolección de datos de control del restaurante sin el sistema, para los tiempos de los pedidos para llevar se registraban las horas de salida de los mismos en sus respectivos tickets, en cuanto a los pedidos para servir, se ingresaban directamente en una tabla de “*Google Sheets*” el número de la mesa, la hora de ingreso y término del pedido. Además de registrar los errores que ocurrían dentro de la cocina en una tabla similar descrita anteriormente.

Calibración de los instrumentos

En cuanto a la prueba en sí, no se empleó ningún instrumento, pues los procesos que requieren medir los tiempos se realizan de manera automática, mientras que para los errores, tampoco necesita de ningún instrumento.

Sin embargo, para la fase de recolección de datos de control, se necesitó de un dispositivo para revisar la hora, requiriendo que tuviese batería y que indicara la misma hora que el dispositivo que imprimía los tickets de venta.

Procedimiento de Recolección de Datos fase B

¿Cómo se recolectó la información?

Al tener los tickets con la fecha de ingreso y término de los pedidos para llevar, al final del día se registraron en la misma tabla que se fueron ingresando las horas de ingreso y término de los pedidos para servir.

En cuanto a los errores, se registraban directamente en las tablas respectivas de datos de control y datos de la prueba en cuanto se encontraban.

Por último, el sistema recolectaba automáticamente el resto de los datos dentro de la base de datos.

Condiciones de observación y/o experimentación

El memorista se encontraba en la caja ingresando los pedidos al sistema mediante la vista TPV en el dispositivo que alojaba el sistema, mientras que los garzones y el personal dentro de la cocina se encargaban de manejar la aplicación dentro de la cocina en caso de que el memorista no pudiese realizarlo.

Pilotaje del proceso

En cuanto al pilotaje de esta fase, simplemente se probó que las métricas se generaran de manera correcta. Para ello, se probó la aplicación en un ambiente controlado sin más estructura definida, se probó ingresando pedidos, esperar unos minutos e indicarle al sistema que estuviese listo y revisando que restara un tiempo parecido al tiempo transcurrido en la vida real.

Reproducibilidad del proceso

El proceso es reproducible, ya que el sistema se encarga automáticamente de la recopilación de datos, teniendo solo que registrar los errores que ocurrirán durante el período que se esté probando. Para recolectar datos de control, se tendría que registrar las fechas de la misma manera descrita anteriormente, mediante los tickets emitidos, o si no emites con la hora, se deberá ingresar los datos de los pedidos en alguna tabla, en vez de esperar hasta el final del día.

4.3. Materiales

¿Qué se necesitó para ejecutar los estudios?

Se necesitó un dispositivo que ejecutara la aplicación, con acceso a internet y algún navegador web. Debido a que la aplicación necesita acceso a internet para cargar algunas librerías utilizadas, por ejemplo *jQuery*. Además de dos dispositivos medianos con acceso a internet y algún navegador web para mostrar la vista de pedido pendientes y la vista de colas de las estaciones.

Condiciones de control y de variación

Control: El hardware utilizado. Se emplearon dos dispositivos que el memorista ya poseía desde antes de empezar el desarrollo de la aplicación. Adicionalmente, se emplearon dos

tabletas idénticas facilitadas por el profesor guía. Las especificaciones del hardware utilizado son:

- ASUS TUF Dash F15 FX517ZR-F15 (lanzado noviembre de 2021, figura 4.1):
 - Pantalla: 15.6", 1920 x 1080 (FHD).
 - CPU: Intel Core i7-12650H, 2.3 GHz.
 - RAM: 16 GB DDR5-SDRAM (4800 MHz).
 - OS: Windows 11 Pro.
- Computador de mesa personalizado (armado enero 2019):
 - Pantalla: 27.0", 1920 x 1080 (FHD).
 - CPU: AMD Ryzen™ 7 3700X, 3.6 GHz.
 - RAM: 32 GB DDR4-SDRAM (3600 MHz).
 - OS: Windows 11 Pro.
- Samsung Galaxy Tab A 8" 2019 SM-T290 (lanzado julio de 2019, figura 4.2):
 - Pantalla: 8", 1280 x 800 (WXGA).
 - CPU: Qualcomm Snapdragon 429 SDM429 (12 nm), Quad-core, 1950 MHz, ARM Cortex-A53, 64-bit.
 - RAM: 2 GB.
 - OS: Android 11.
- iPad (6th generation) (lanzado marzo 2018, figura 4.3):
 - Pantalla: 9.7", 1536 x 2048 (4:3).
 - CPU: Quad-core 2.34 GHz (2 x Hurricane + 2 x Zephyr).
 - RAM: 2 GB.
 - OS: iPadOS 16.3.1.



Figura 4.1: ASUS TUF Dash F15 (2022). Fuente: <https://www.asus.com>

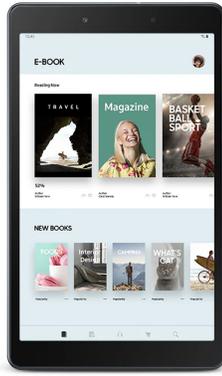


Figura 4.2: Samsung Galaxy Tab A SM-T290 (2019). Fuente: <https://www.samsung.com>



Figura 4.3: iPad (6th generation) (2018). Fuente: <https://support.apple.com>

Variación: No hay, pues todos ocuparon la misma aplicación.

4.4. Reproducibilidad del estudio

El hardware está especificado en el apartado de condiciones de control. El código fuente del software se encuentra en un repositorio de *GitHub*¹. Con el código fuente se puede ejecutar la aplicación tras instalar las librerías necesarias con el comando `pip install -r requirements.txt` o instalar las librerías de `requirements.txt` manualmente. Debido a que el programa es personalizado, depende de archivos externos que no se encuentran en el repositorio, actualmente está conectado a un *Google Sheets*, donde la estructura es la siguiente:

¹<https://github.com/jingluoz/KitchenPOS> (26 de junio de 2023)

Tabla 4.29: Hoja 1: “artículos”.

Columna	Descripción
1.Familia	Familia del artículo.
2.Estación	Tipo de estación de preparación del artículo.
3.Nombre	Nombre del artículo.
4.Descripción	Descripción del artículo.
5.Precio	Precio del artículo.
6.Cantidad_ Utilizada	Cantidad utilizada del artículo.
7.Es_ Combinación	Casilla de verificación, TRUE si el artículo es una combinación, FALSE en caso opuesto.
8.Es_ Visible	Casilla de verificación, TRUE si el artículo es un artículo visible dentro del sistema TPV, FALSE en caso opuesto.
9.Es_ Editable	Casilla de verificación, TRUE si el artículo es modificable en caso de ser una combinación, FALSE en caso opuesto.

Tabla 4.30: Hoja 2: “combinaciones”.

Columna	Descripción
1.Artículo	Nombre del artículo principal.
2.Sub-artículo	Nombre del sub-artículo que conforma al artículo.
3.Cantidad	Cantidad del sub-artículo.

Tabla 4.31: Hoja 3: “notas”.

Columna	Descripción
1.Nombre	Nombre del tipo de nota.
2.Precio	Valor agregado de la nota.
3.Tipo	Tipo de notas, actualmente sólo están los tipos “Cantidad” y “Booleano”.

Donde la hoja 1(4.29) representa a todos los artículos de la carta, hoja 2 (4.30) para las combinaciones de los artículos de la carta y la hoja 3 (4.31) para las notas de los artículos.

Además, se debe crear una base de datos en *Postgres* en el dispositivo que ejecuta la aplicación.

Por último, para la configuración de la aplicación es necesario crear un archivo llamado *variables.py* con una variable *ips* una lista con la dirección IP de la red local, *db_password* y *db_name* la clave y nombre de la base de datos creada en *Postgres* y las variables *sample_spreadsheet_id*, *client_id*, *client_secret* y *refresh_token* del archivo *Google Sheets* (se pueden obtener configurando el proyecto desde el servicio *Google Cloud*²).

²<https://console.cloud.google.com> (26 de junio de 2023)

Capítulo 5

Resultados

En este capítulo se presentan los resultados obtenidos de los procesos de experimentación en general. Todos los gráficos mostrados, tanto en este apartado como en el anexo, se realizaron en el software *RStudio*¹.

5.1. Resultados del experimento de la vista TPV

En esta sección, se mostrarán los resultados del experimento realizado sobre la vista TPV.

5.1.1. Wilson y Wixon [4]

Los resultados a comparar son algunas de las métricas de Wilson y Wixon [4], que se definen de la siguiente manera:

- Número de errores: esta métrica se refiere a la cantidad de errores cometidos durante el desarrollo de una tarea. Para analizar la cantidad de errores, se promediaron los resultados para cada usuario durante las 7 tareas desarrolladas.
- Número de ayudas necesarias: esta métrica se refiere a la cantidad de ayudas solicitadas por el usuario al experimentador durante el desarrollo de la tarea. Para analizarla, se promedió el total de ayudas requeridas por usuario durante las 7 tareas.
- Número de usuarios que cometen el mismo error: esta métrica se refiere a los usuarios que cometieron el error al menos una vez durante las 7 tareas. Se registró el recuento de estos errores en general, para analizar cuáles fueron más recurrentes en la muestra total.
- Número de usuarios que completan la tarea con éxito: esta métrica se refiere a la cantidad de usuarios que lograron completar el objetivo planteado en la tarea. En este caso particular, se sumó la cantidad de tareas completadas para cada usuario, y finalmente se reportó la proporción entre las tareas en las que se cumplió el objetivo y el total de estas.

¹<https://www.rstudio.com/tags/website/> (22 de julio de 2023)

Número de errores

Con respecto al número promedio de errores agregados para todas las tareas, los usuarios obtuvieron un mínimo de 0, una mediana de 0.47, un promedio de 0.43 y un máximo de 0.85. Los datos del promedio de la cantidad de errores cometidos por la muestra se observan en las figuras del anexo B.1 y B.2.

Número de ayudas necesarias

Con respecto al promedio de ayudas necesarias agregadas para todas las tareas, los usuarios obtuvieron un mínimo de 0, una mediana de 0.47, un promedio de 0.43 y un máximo de 0.85. Los datos del promedio de la cantidad de ayudas proporcionadas se observan en las figuras del anexo B.3 y B.4.

Número de usuarios que cometen el mismo error

Durante el desarrollo de las tareas por parte de los usuarios, se observaron 6 errores distintos. Los errores más cometidos fueron que los usuarios no sabían que para cambiar un componente de las combinaciones había que abrir las notas y que para seleccionar el nuevo artículo debían seleccionar los artículos desde el menú de artículos, ocurriendo 9 veces cada uno. El segundo error más cometido es el de “No encontraba el botón de notas”, el cual fue cometido al menos una vez por 8 usuarios, lo que significa que gran parte de los usuarios no pudieron identificar el símbolo designado a las notas o bien no sabían dónde se encontraba. El tercer error más cometido por 4 usuarios distintos fue el de “No encontraba el botón de Verduras”. Esto se debe a que, para poder presionar la familia de Verduras, es necesario desplazarse mediante un scroll horizontal en la sección de las familias, y los usuarios que no se dieron cuenta de esto. El cuarto error más recurrente, con 3 ocurrencias, fue que los usuarios no supieron localizar el botón de nuevo pedido, se cree que estos errores fueron ocasionados por motivos similares al del error anterior, pues para acceder a él, se necesita pasar el cursor en la barra lateral izquierda para poder visualizar el menú de navegación. Por último, el error que menos ocurrió fue el de que al eliminar un elemento de la lista de pedidos, se presionó el botón de eliminar todo el pedido en vez de eliminar un único artículo.

En la siguiente tabla se puede visualizar de forma resumida los errores cometidos por los usuarios.

Tabla 5.1: Diferentes errores cometidos u observados en la experiencia y frecuencia de usuarios.

Error cometido	Frecuencia
No sabía que para cambiar un artículo había que abrir las notas	9
No sabía que se tenía que seleccionar el nuevo artículo en la carta	9
No encontraba el botón de notas	8
No encontraba el botón de Verduras	4
No encontraba el botón de Nuevo pedido	3
Presiona botón de “Eliminar” todo el pedido en vez de un artículo	2

Número de usuarios que completan la tarea con éxito

En este apartado, todos los usuarios completaron las 7 tareas planteadas. Esto representa un 100 % de éxito.

5.1.2. Encuesta Post-Experiencia

De los resultados de la encuesta realizada a los usuarios tras realizar el experimento, se le dio un enfoque a la dificultad percibida por parte de los usuarios:

Las tareas más difíciles

Para esta métrica simplemente se contó cuantas veces una tarea era percibida como difícil por los usuarios, generando la siguiente tabla:

Tabla 5.2: Tareas y las veces que fueron percibidas como difíciles.

Tarea	Frecuencia
Tarea 1	0
Tarea 2	0
Tarea 3	1
Tarea 4	6
Tarea 5	9
Tarea 6	3
Tarea 7	1

Las dos tareas percibidas como las más complejas son las tareas 4 y 5, mencionadas por 6 y 8 usuarios respectivamente, siendo ambas tareas las únicas relacionadas con el uso del sistema de notas.

Dificultad de la experiencia

Para esta métrica se tomaron las respuestas proporcionadas por los usuarios para realizar un análisis numérico, en una escala del 1 al 5, siendo 1 fácil y 5 difícil, donde se obtuvieron un mínimo de 1, una mediana de 2, una media de 2.4 y un máximo de 4.

Sugerencias de los usuarios

Aunque esta no es una métrica como tal, los comentarios ayudarán a guiar los siguientes pasos del prototipo de la aplicación. Para esto, se tomaron todas las sugerencias de los usuarios y se creó una lista con las diferentes sugerencias, agrupando aquellas que fueran similares o relacionadas en una única sugerencia, produciendo la siguiente lista:

- Añadir un ajuste del tamaño de la fuente de la vista.
- Modificar el sistema de cambios por un modal que muestre los artículos por los que se puede cambiar un artículo.
- Modificar el botón de notas para que sea más reconocible.
- Emplear botones para desplazarse por las secciones de familias y artículos.

- Buscador de artículos por algún parámetro.

Estos cambios son solo propuestas y no son determinantes para el futuro de la aplicación; sin embargo, se tendrán en cuenta en caso de continuar con el proyecto.

5.2. Resultados del experimento de vistas de pedidos en ambiente controlado

En esta sección, se mostrarán los resultados del experimento realizado sobre las vistas relacionadas con los pedidos en un ambiente controlado.

5.2.1. Pruebas controladas

Una vez que se repitió 5 veces cada prueba definida anteriormente, los resultados se reflejan en la siguiente tabla:

Tabla 5.3: Pruebas definidas para las vistas de pedidos y las veces que pasaron

Prueba	Frecuencia
Prueba 1	5
Prueba 2	5
Prueba 3	5
Prueba 4	5
Prueba 5	5
Prueba 6	5
Prueba 7	5
Prueba 8	5
Prueba 9	5

Esto indica que, al menos bajo las condiciones definidas, con los valores iniciales definidos anteriormente en la sección de Evaluación 4.2.2, el sistema se comporta como se esperaba, sin errores en ninguna de las pruebas, es decir, tiene una robustez del 100% bajo estas condiciones.

5.3. Resultados del experimento de vistas de pedidos en ambiente real

En esta sección, se mostrarán los resultados del experimento realizado sobre las vistas relacionadas con los pedidos en un ambiente real, presentando los resultados obtenidos tanto de los datos del restaurante con el sistema instalado, como los datos del restaurante sin el sistema. Las métricas utilizadas son las siguientes:

- Número de pedidos: esta métrica se refiere al total de pedidos por día. Para analizar la cantidad de pedidos, se agruparon por día y se contaron cuántos pedidos hubo por día.

- Tiempo transcurrido: esta métrica se refiere al tiempo en segundos que transcurre desde que se ingresa un pedido hasta que está listo. Para analizar el tiempo, se toma el tiempo registrado para hacer un análisis de todos los datos y una agrupación por día para comparar la métrica en el transcurso de los días.
- Número de artículos: esta métrica se refiere al número de artículos de los pedidos. Para analizar la cantidad de artículos, se toma el promedio de los artículos por pedidos.
- Número de artículos diferentes: esta métrica se refiere al promedio de artículos diferentes de los pedidos. Para analizar la cantidad de artículos diferentes, se calcula el promedio de los artículos por pedido.
- Segundos por artículo: esta métrica se refiere a la cantidad de segundos que toma preparar un pedido distribuido por la cantidad de total artículos del mismo pedido. Para analizar esta métrica, se promedian de la misma para comparar ambas muestras.
- Segundos por artículo diferentes: esta métrica se refiere a la cantidad de segundos que toma preparar un pedido distribuido por la cantidad de artículos diferentes del mismo pedido. Para analizar esta métrica, se promedian de la misma para comparar ambas muestras.
- Cantidad de pedidos previos en cola: esta métrica se refiere a la cantidad de pedidos que estuvieran en la cola de pedidos pendientes a la hora de ingresar un pedido. Para analizar esta métrica, se promediarán la cantidad de pedidos que estuvieran en la cola.
- Número de errores: esta métrica se refiere a la cantidad de errores cometidos durante el período de recopilación de datos. Para analizar la cantidad de errores, se compararán los errores en sí y el número de veces que ocurrió.

5.3.1. Datos recopilados en el restaurante con el sistema

En esta sección se presentan los datos recopilados después de 8 días de prueba del sistema en el restaurante de comida china.

Número de pedidos

Con respecto a la cantidad de pedidos, hubo un total de 149 pedidos, el mínimo por día fue de 10, la media fue de 18.63 y el máximo de 31. Ninguno de los 149 pedidos fue programado. Estos datos se pueden observar graficados en las figuras del anexo B.17 y B.18.

Tiempo transcurrido

Con respecto a los tiempos que transcurrían desde que un pedido era ingresado hasta ser completado, hubo un mínimo de 89 segundos, una media de 953.26 segundos, una mediana de 990 segundos y un máximo de 1848 segundos en la preparación de los pedidos. Estos datos se pueden observar graficados en la figura del anexo B.20.

Número de artículos

Con respecto a la cantidad de artículos de un pedido, la mínima fue de 1, la media fue de 6.01, la mediana de 6 y la máxima de 10. Estos datos se pueden observar graficados en la figura del anexo B.23.

Número de artículos diferentes

Con respecto al número de diferentes artículos de un mismo pedido, la mínima fue de 1, la media de 3.46, la mediana de 4 y la máxima de 7. Estos datos se pueden observar graficados en la figura del anexo B.25.

Segundos por artículo

Con respecto al tiempo transcurrido de un pedido distribuido en la cantidad de artículos totales de cada pedido, el mínimo fue de 74.5, la media de 167.08, la mediana de 160.75 y la máxima de 355. Estos datos se pueden observar graficados en la figura del anexo B.26.

Segundos por artículo diferentes

Con respecto al tiempo transcurrido de un pedido distribuido en la cantidad de artículos diferentes de cada pedido, el mínimo fue de 89, la media de 287.29, la mediana de 275.5 y la máxima de 593.5. Estos datos se pueden observar graficados en la figura del anexo B.27.

Cantidad de pedidos previos en cola

Con respecto a la cantidad de pedidos que estuvieran en la cola de pedidos a la hora de ingresar un nuevo pedido, el mínimo fue de 0, la media de 1.89, la mediana de 2 y el máximo de 5. Estos datos se pueden observar graficados en la figura del anexo B.28.

Número de errores

En el mismo período, se identificaron dos categorías de errores. La primera categoría incluye los errores generados directamente por el sistema. Un ejemplo de esto es la representación incorrecta de los colores que indican el estado de un artículo. La segunda categoría comprende errores causados por la utilización incorrecta del sistema por parte de los usuarios. Un ejemplo de esto sería un usuario que introduce incorrectamente la cantidad de un artículo preparado, dando como resultado un artículo excedente y causando que la próxima vez que se introduzca el artículo se genere un error al mostrar la cantidad correcta a preparar. La frecuencia de estos errores detectados durante este período se resume en la siguiente tabla:

Tabla 5.4: Errores ocurridos en el restaurante con el sistema y las veces que pasaron

Error	Tipo	Frecuencia
Los dispositivos dentro de la cocina se bloqueaban.	Sistema	32
Color de artículo equivocado	Sistema	20
El artículo solicitado no se encuentra en el sistema	Sistema	18
Muestra cantidad errónea de cantidad a preparar	Sistema	10
Aparece botón "Listo" de un pedido listo múltiple veces	Sistema	1
Se ingresó la cantidad de un pedido de forma equivocada en la vista dentro de la cocina	Usuarios	10
Se ingresó de forma errónea el pedido desde la vista TPV	Usuarios	2

La mayoría de los errores identificados se originaron directamente en el sistema. El error más frecuente, con 32 ocurrencias, fue que los dispositivos se bloqueaban automáticamente

después de un período de inactividad. Este problema no se pudo solucionar debido a las limitaciones de las tabletas utilizadas.

El segundo error más recurrente, ocurriendo 20 veces, fue que los artículos se mostraban con un color incorrecto.

Un tercer problema, aunque más bien un inconveniente que un error, fue que, durante el período de prueba del sistema, hubo clientes que pedían alimentos que no estaban en el menú del restaurante (el cual sirve como base para el sistema). Aunque estos alimentos se podían pedir, el sistema actual no permite agregar un artículo que no esté previamente cargado en la base de datos. Este problema se presentó 18 veces.

Los errores con la cuarta frecuencia más alta, con 10 ocurrencias cada uno, incluyen: el sistema mostrando una cantidad incorrecta de artículos a preparar en la vista de colas, y los usuarios ingresando una cantidad incorrecta en la vista de pedidos.

Se registró un error en dos ocasiones donde el encargado de tomar los pedidos ingresó un pedido de forma incorrecta en la vista TPV.

Por último, un error que ocurrió solamente una vez se debió a un problema en el código que agregaba múltiples botones de “Listo” en los pedidos.

5.3.2. Datos recopilados en el restaurante sin el sistema

En esta sección, se describen los datos recolectados durante 6 días de prueba en un restaurante de comida china, donde no se utilizó el sistema desarrollado.

Número de pedidos

Con respecto a la cantidad de pedidos, hubo un total de 93 pedidos, el mínimo por día fue de 12, la media fue de 15.5 y el máximo de 25. Estos datos se pueden observar graficados en las figuras del anexo B.38 y B.39.

Tiempo transcurrido

Con respecto a los tiempos que transcurrían desde que un pedido era ingresado hasta ser completado, hubo un mínimo de 166 segundos, una media de 803.6 segundos, una mediana de 798 segundos y un máximo de 1728 segundos en la preparación de los pedidos. Estos datos se pueden observar graficados en la figura del anexo B.41.

Número de artículos

Con respecto a la cantidad de artículos de un pedido, la mínima fue de 1, la media fue de 5.04, la mediana de 5 y la máxima de 10. Estos datos se pueden observar graficados en la figura del anexo B.44.

Número de artículos diferentes

Con respecto al número de diferentes artículos de un mismo pedido, la mínima fue de 1, la media de 3.17, la mediana de 3 y la máxima de 7. Estos datos se pueden observar graficados en la figura del anexo B.46.

Segundos por artículo

Con respecto al tiempo transcurrido de un pedido distribuido en la cantidad de artículos totales de cada pedido, el mínimo fue de 83, la media de 170.04, la mediana de 158 y la máxima de 355. Estos datos se pueden observar graficados en la figura del anexo B.47.

Segundos por artículo diferentes

Con respecto al tiempo transcurrido de un pedido distribuido en la cantidad de artículos diferentes de cada pedido, el mínimo fue de 139.5, la media de 265.97, la mediana de 248.5 y la máxima de 593.5. Estos datos se pueden observar graficados en la figura del anexo B.48.

Cantidad de pedidos previos en cola

Con respecto a la cantidad de pedidos que estuvieran en la cola de pedidos a la hora de ingresar un nuevo pedido, el mínimo fue de 0, la media de 0.55, la mediana de 0 y el máximo de 4. Estos datos se pueden observar graficados en la figura del anexo B.49.

Número de errores

Durante este período, se identificaron varios errores, que se resumen en la siguiente tabla:

Tabla 5.5: Errores ocurridos en el restaurante sin el sistema y las veces que pasaron

Error	Frecuencia
Antes de entregar el pedido, faltaban artículos.	2
Se colocó un artículo en el pedido incorrecto.	2
No se llevó el ticket del pedido a la cocina.	1
Se entregó un pedido al cliente equivocado.	1
Se ingresó de forma errónea el pedido desde la aplicación TPV.	1

Los errores más frecuentes fueron la falta de artículos en el pedido antes de la entrega y la colocación de un artículo en el pedido incorrecto, ambos con una frecuencia de 2. Los errores menos frecuentes fueron el ingreso erróneo del pedido en la aplicación TPV, la entrega del pedido al cliente equivocado, y el olvido de llevar el ticket del pedido a la cocina, cada uno de estos errores ocurrió una sola vez.

Capítulo 6

Análisis y discusión de resultados

En este capítulo se analizarán los resultados de los tres experimentos descritos en el capítulo de “Evaluación”.

6.1. Análisis de resultados del experimento 1

En primer lugar, se analizarán los resultados obtenidos del primer experimento, centrado en los errores cometidos en las métricas de Wilson y Wixon [4]. En la tarea 5, que corresponde a realizar un cambio en una combinación, se observó que 9 de los 10 usuarios de la muestra cometieron errores. Esto sugiere que la mayoría de los usuarios, a excepción de uno, no comprendieron que, para poder acceder al menú de modificación de una combinación, deben acceder a las notas. Este patrón indica que la forma en que se manejan los cambios dentro del sistema actual podría necesitar una revisión para ser más intuitiva. Sin embargo, como se espera que el programa se utilice diariamente en un restaurante, es posible que los usuarios se acostumbren al método actual con el tiempo.

Por otro lado, el método para agregar un artículo desde el menú de artículos al realizar un cambio de artículos de una combinación, es similar al de la aplicación en la que se basó el desarrollo inicial de la vista (*Itactil*). Una posible mejora sería crear una interfaz propia para los cambios, lo que podría dar como resultado un flujo de trabajo más intuitivo, fluido y directo.

Al analizar más a fondo las métricas de Wilson y Wixon [4], se puede concluir que después de ayudar a los usuarios con un error específico, estos no volvieron a cometer el mismo error. Esto indica que la curva de aprendizaje para la vista es relativamente baja. La mayoría de los errores cometidos por los usuarios se produjeron porque no se les había enseñado cómo funcionaba el sistema de antemano. El resto de los errores fueron cometidos por no saber dónde se encontraban los elementos en la vista para poder realizar una acción.

Además, la encuesta realizada a los usuarios después de la experiencia reforzó la necesidad de modificar el flujo de cambios de artículos, ya que fue la tarea más votada como difícil.

En cuanto a las sugerencias de los usuarios, se diseñaron los siguientes bosquejos en donde

se aplican estos cambios:

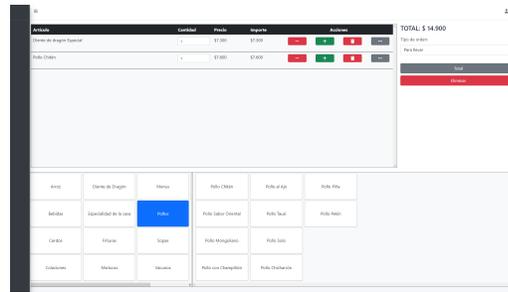
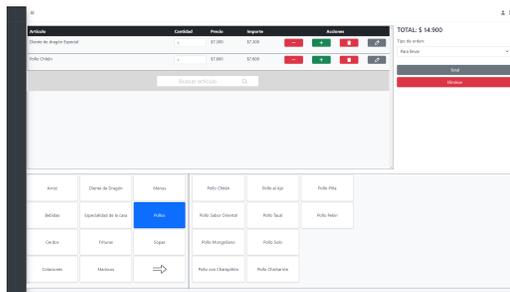
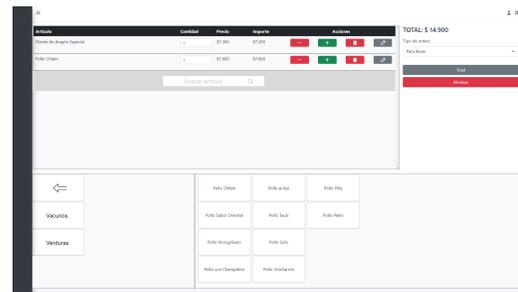


Figura 6.1: Vista sistema actual.



(a) Vista página 1.



(b) Vista página 2.

Figura 6.2: Vista TPV con cambios propuestos de reemplazo de scroll por flechas, cambio de icono de notas y barra de búsqueda de artículos.



Figura 6.3: Vista propuesta de configuración de fuentes de las vistas de la aplicación.

Donde en las figuras 6.2a y 6.2b, se aplicaron los cambios en el modo de desplazamiento por los menús de artículos y familias, pasando de un scroll horizontal a un desplazamiento mediante botones de siguiente y anterior. Además, se realizó un cambio en el icono de notas, que pasó de ser simplemente “...” a un lápiz, que comúnmente se usa para simbolizar una acción de edición o modificación. También se añadió una barra de búsqueda para los artículos en el espacio que se ocuparía si se fuera a agregar un artículo. Por otro lado, la imagen de la figura 6.3 surge como sugerencia de algunos usuarios que tenían dificultades para leer debido al tamaño de la letra. Al crear una vista para la configuración de las vistas, se permite no solo cambiar las fuentes de la aplicación, sino que también en el futuro permitiría la configuración de los colores que se emplean en las vistas.

Finalmente, el flujo de cambios o modificación de combinaciones podría quedar de la siguiente manera:

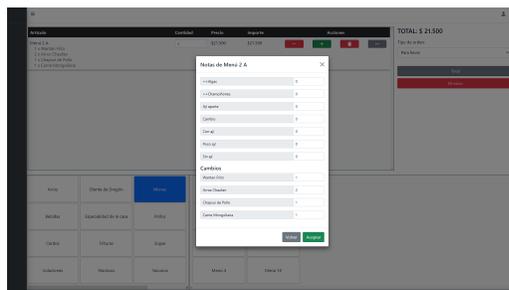
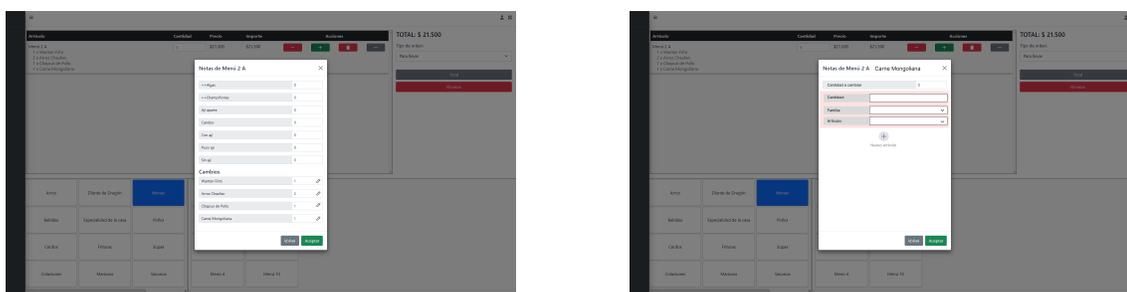


Figura 6.4: Vista sistema actual.



(a) Vista notas propuesto.

(b) Vista cambio de artículos.

Figura 6.5: Vista TPV con cambio del flujo de modificación de combinaciones.

Las principales diferencias entre el flujo actual y el flujo propuesto radican en cómo se realiza un cambio. En la vista actual, para realizar un cambio es necesario modificar el número al lado de un artículo en la sección de cambios que se encuentra en la parte inferior de la vista 6.4, y luego se debe salir del modal para seleccionar un nuevo artículo. En cambio, en el nuevo flujo se propone que para realizar un cambio, es necesario presionar el botón de editar que está al lado de los artículos de cambio en la figura 6.5a. Esto abriría un nuevo modal (figura 6.5b) con los datos requeridos para los cambios, en donde uno puede ingresar la cantidad del artículo que se quiera cambiar, datos del artículo por el que se quiera cambiar y añadir más artículos en caso de que uno quisiera cambiar el artículo por más de un artículo diferente. Por ejemplo, en un “Menú para 2A” vienen 2 “Arroz Chaufan” y un cliente quisiera cambiar ambos arroces por un “Arroz Blanco” y un “Arroz Chaufan Grande”.

6.2. Análisis de resultados del experimento 2

Este experimento fue sobre todo una verificación de la robustez del prototipo para asegurarse de que la aplicación funcionara como se esperaba antes de probar el sistema en casos reales, para minimizar los errores. Por tanto, la única conclusión que se puede sacar de los resultados de este experimento es que, bajo las condiciones iniciales, la aplicación se comporta como debería.

Como sugerencia para futuros estudios o desarrollos similares, esta fase debería realizarse utilizando múltiples dispositivos. Durante esta fase de prueba, solo se probó con una tableta de la figura 4.3 que tenía el memorista, en la cual no se detectó ningún error. Sin embargo, durante la realización del tercer experimento, se utilizaron las tabletas Samsung de la figura 4.2, en las cuales ocurrió un error que no afectaba la lógica del sistema, pero sí afectaba la forma en que se presentaban los pedidos, ya que a veces se mostraban con un color erróneo.

6.3. Análisis de resultados del experimento 3

El objetivo de este experimento es evaluar si la aplicación efectivamente apoya el trabajo dentro de la cocina. Para ello, primero se comparará la cantidad de artículos entre ambos casos para verificar si las pruebas son comparables. Si las muestras son muy pequeñas, podrían no ser representativas. Sin embargo, dado que el caso de las pruebas sin sistema es casi de 100, mientras que las del caso con sistema llegan a casi 150, en ambos casos la cantidad no es menor. Tal vez hubiera sido mejor continuar con la recopilación de datos en el ambiente sin sistema hasta llegar a obtener una cantidad de pedidos similar a la del caso con sistema, sin embargo, por restricciones de tiempo, no se pudo realizar de este modo.

Luego, en vez de comparar directamente el tiempo transcurrido desde que se ingresa un pedido hasta el término de la preparación, es mejor comparar las métricas de “Segundos por artículo” y “Segundos por artículo diferentes”, ya que en el primer caso, el tiempo transcurrido puede variar demasiado dependiendo de la cantidad de artículos que componen un pedido. La media de segundos por artículo para los datos con sistema fue de 167.08, mientras que la media para los datos sin sistema fue de 170.04, siendo números muy similares. Por otro lado, la media de segundos por artículos diferentes en los datos del ambiente con sistema fue de 287.29 segundos, mientras que la media para los datos sin sistema fue de 265.97. Aunque los resultados de la comparación no son muy concluyentes debido a la similitud de los resultados, al incluir la métrica de “Cantidad de pedidos previos en cola”, se tiene que la media para los datos con sistema es de 1.31, mientras que la media para los datos sin sistema es de 0.55. Esto indica que durante el período de experimentación con sistema la mayor parte del tiempo había un pedido en cola a la hora de agregar un pedido, mientras que en el caso sin sistema había un pedido en cola en 1 de cada 2 pedidos. Esto sugiere que, aunque haya una mayor cantidad de pedidos en cola, no se refleja un aumento en el tiempo de preparación en los resultados de las métricas anteriores. Por lo tanto, se puede inferir que o el sistema aporta en menor medida en la preparación de alimentos, o bien se omitieron más métricas para realizar la comparación.

Al comparar los errores de ambos casos, los errores que se presentaron durante el período de prueba con sistema, la mayoría de los errores no presentan un problema mayor. El principal error fue ocasionado por los dispositivos dentro de la cocina, ya que no se podía desactivar el bloqueo de pantalla en los dispositivos, estos errores se solucionaron desbloqueando los dispositivos cuando ocurriera.

El segundo error más cometido fue que no se actualizaban los colores de los estados de los artículos, esto no tenía una repercusión muy grande, ya que al revisar la base de datos, los estados se cambiaban de manera correcta, indicando que era un bug visual. Sin embargo, al probar nuevamente el sistema con otros dispositivos, el error sólo ocurría en los dispositivos

utilizados para la prueba.

El tercer error / inconveniente fue que había pedidos que no se encontraran en la carta del restaurante. Esto se podría solucionar agregando una interfaz que permita crear un artículo desde la vista de toma de pedidos, ya que en el procedimiento actual del restaurante en el que se probó el sistema, simplemente se cambia el nombre de un artículo. Sin embargo, en el sistema desarrollado, un artículo no es meramente un nombre, sino que al menos se necesita saber en qué estación se necesita encolar y la cantidad de espacio que se requiere en dicha estación. Por ello, para solucionar este inconveniente se propone modificar la interfaz TPV de la siguiente manera:

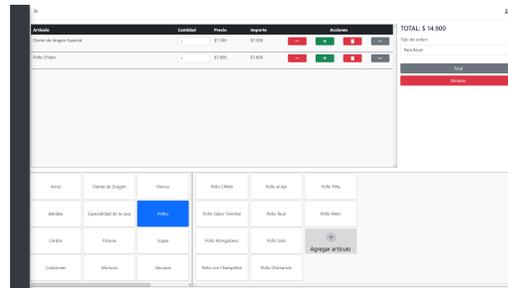
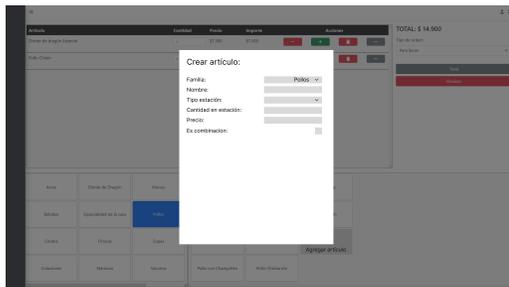
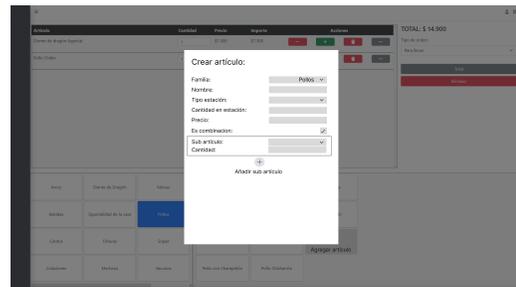


Figura 6.6: Vista propuesta de TPV.



(a) Vista creación artículo no combinación.

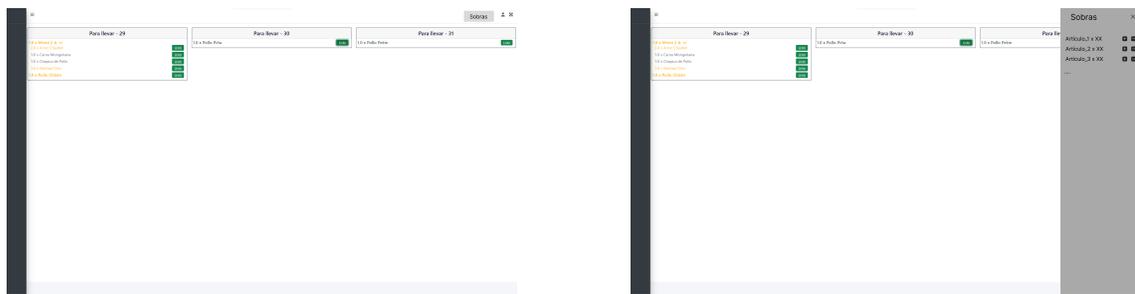


(b) Vista creación artículo combinación.

Figura 6.7: Vistas TPV propuesta creación artículos tras presionar el botón de añadir artículo.

En estas propuestas, se añade un botón de “Agregar artículo” en la figura 6.6 para poder acceder a las vistas de las figuras 6.7. Estas vistas presentan una forma simple de agregar artículos al sistema, con los datos mínimos necesarios para que el sistema funcione.

Los siguientes errores que se presentaron en el sistema durante las pruebas, fueron los de que el sistema “Muestra cantidad errónea de cantidad a preparar” y “Se ingresó la cantidad de un pedido de forma equivocada en la vista dentro de la cocina”, una consecuencia directa de que un usuario haya ingresado una cantidad incorrecta en la vista de pedidos, es que el sistema muestre una cantidad errónea a preparar. Al tener la misma frecuencia, esto indica que son el mismo error. Este error pudo ocurrir debido a que el sistema no muestra las sobras del sistema y tampoco hay forma de gestionarlas, por ello se propone el siguiente cambio en la interfaz de pedidos:



(a) Vista de pedidos con botón de sobras. (b) Vista de pedidos con el manejo de sobras.

Figura 6.8: Vistas de pedidos propuesta para el manejo de sobras.

En la figura 6.8a se añadió un botón en la barra superior para que no interfiera con los pedidos de la vista. Luego, en la figura 6.8b se añadió una sección para poder aumentar o reducir la cantidad de una sobra.

En cuanto al resto de errores que se encontraron en esta fase, está el error en el que se ingresó el pedido de forma errónea desde la vista TPV, siendo este el único error que ocurrió en ambas pruebas. Sin embargo, esto puede ocurrir en cualquier caso, ya que es un error que tiene que ver con el usuario en sí. Una funcionalidad que se puede aplicar a un sistema para reducir estos errores puede ser que, al mostrar el total, se muestre un resumen del pedido (funcionalidad que se implementó desde el inicio del desarrollo). Pero si el usuario no revisa el resumen por algún motivo, no se puede hacer mucho más. Finalmente, el error en el que aparecían múltiples botones de “Listo” en la vista, se solucionó de forma inmediata cuando se detectó.

Como no se registraron errores en cuanto a que los usuarios no hayan entendido cuál era el siguiente platillo o qué artículos faltaban en cada pedido, se puede interpretar que las vistas diseñadas son claras en cuanto a la información que muestran y que con una capacitación al principio de la experimentación, bastó para que los usuarios pudieran emplear las vistas de la forma correcta.

Por otro lado, los errores que se presentaron en la segunda fase fueron problemas de los usuarios que tienen que ver más que nada con errores de ensamblado de pedidos y errores que ocurrieron por no saber el estado de los artículos de los pedidos. La mayoría de los errores no ocurrieron en la situación con el sistema, de lo que se puede concluir que, o bien, el sistema ayuda a solucionar estos problemas o simplemente no se detectaron y falta probar más el sistema.

6.4. Posibles extensiones del sistema

Aunque las pruebas se realizaron pensando en un restaurante de comida china, eso no significa que el programa no funcione con otros tipos de restaurantes. Lo único que liga al programa con este tipo de comida es meramente la lista de artículos y las estaciones de trabajo. El sistema fue probado con datos de restaurantes de comida china simplemente por el contexto en el que se encontraba el memorista, ya que creció en uno, conociendo a varias personas que han trabajado en diferentes restaurantes a lo largo de su vida, lo que facilitó

en gran medida la comprensión del funcionamiento y la lógica que siguen los restaurantes de este sector.

El sistema actual puede ser instalado en un restaurante de cualquier tipo de comida, simplemente bastaría con cambiar las tablas descritas en 4.4. Se tendría que agregar los artículos de la carta del menú del restaurante donde se quiera instalar, identificar las combinaciones que tengan dentro de la misma, agregando las combinaciones encontradas en una tabla y agregar las posibles notas o extras que se quieran. Luego, el sistema se encarga de crear una estación de cada tipo con una capacidad predeterminada. Si se quisiera modificar o crear más estaciones, se tendría que crear o modificar vía la interfaz de manejo de base de datos que ofrece *Django*. Por último, bastaría con agregar las prioridades que tienen las familias.

El único inconveniente que se tiene con este proceso, es que las prioridades dependen únicamente de la familia de un artículo y el tipo de pedido. Sin embargo, esto se podría modificar agregando un campo de artículo y modificando los campos de dicha tabla para que sean opcionales para poder dar mayor libertad en el manejo de prioridad dentro de los restaurantes. Esta solución fue considerada una vez se empezaron las pruebas, siendo muy tarde para modificar algo tan importante como las prioridades.

Por último, hablando con diferentes personas sobre el sistema, se pensó que el sistema no sólo sirve para este tipo de servicios. A fin de cuentas, es un programa que permite ingresar un pedido, una vista que muestra los pedidos con sus estados y una vista que muestra el orden de preparación de elementos. Esto podría encajar con un flujo similar al de las tiendas en línea con retiro en el local, donde los usuarios ingresan los pedidos desde la vista TPV en la página de la tienda, los distintos pedidos son mostrados en las tiendas y las estaciones podrían ser pasillos y la prioridad podría ser dependiente de la ubicación de un artículo dentro de la bodega. Para esto, se tendría que agregar algunos datos del cliente, cambiar los datos del cliente a obligatorios, quitar los tipos de pedidos, las mesas y que los pedidos puedan ser programables.

Capítulo 7

Conclusión y Trabajo Futuro

La gestión de un restaurante puede ser un desafío complejo, no obstante, la adopción de diversas tecnologías puede simplificar este proceso considerablemente. Entre estas tecnologías, los Terminales de Punto de Venta (TPV), también conocidos por su término en inglés “Point Of Sale”, destacan por su uso frecuente en este tipo de negocios. Actualmente, existe un amplio abanico de estos sistemas disponibles en el mercado, cada uno con sus funcionalidades distintivas. Algunos de estos sistemas permiten ingresar pedidos directamente desde un dispositivo electrónico. Una vez los pedidos son registrados, algunos sistemas optan por imprimir un ticket con la orden detallada, mientras que otros prefieren visualizar la información de los pedidos en pantallas ubicadas dentro de la cocina del establecimiento.

Sin embargo, de todas las aplicaciones y sistemas revisadas, ninguna posee un sistema de colas que muestre un orden de preparación según prioridades para ayudar a la hora de tomar decisiones para la preparación de pedidos, optimizando el tiempo de preparación de los artículos.

Como resultado de este trabajo de título, se ha desarrollado una aplicación TPV multifuncional que cumple con las operaciones básicas de gestión de pedidos en un restaurante. Esta aplicación, probada en el sistema operativo Windows para la interfaz TPV y Android para la visualización de pedidos y colas de preparación, se encarga de la recepción de pedidos, la modificación y gestión de los mismos.

La interfaz de registro de pedidos de este sistema se inspiró y modeló sobre la base de una aplicación existente conocida como *Itactil*. Por otra parte, el diseño de las vistas restantes se orientó bajo la premisa de proporcionar sólo la información necesaria, enfocándose en la simplicidad y la facilidad de uso. El desarrollo de estas vistas se realizó mediante un proceso iterativo, apoyado en reuniones semanales con el profesor guía y compañeros de la carrera, evolucionando desde pantallas informativas hasta la versión final que vemos hoy.

El prototipo desarrollado resuelve el problema de negocio planteado, proporcionando una representación visual efectiva de los estados de los artículos, mediante el empleo de colores. Adicionalmente, con el objetivo de optimizar la preparación de los pedidos, se incorporó una interfaz que exhibe los artículos agrupados y encolados según su tipo y estación de prepa-

ración, siguiendo un criterio de orden de salida que se rige por múltiples parámetros. Esto incluye el orden de llegada de los pedidos y las prioridades asignadas a diferentes artículos, entre otros parámetros.

La evaluación del sistema TPV desarrollado se hizo puntualmente, usando como escenario un restaurante de comida china. Esta elección se basó en la experiencia previa del memorista en este tipo de establecimientos, lo que permitió una mayor comprensión del contexto y las necesidades específicas de los usuarios y expertos de dominio. Aunque el tiempo disponible para la prueba del sistema fue limitado debido a los plazos establecidos en el marco de la memoria, se implementaron pruebas exhaustivas para garantizar el correcto funcionamiento y minimizar cualquier impacto negativo en el negocio del restaurante.

Los resultados obtenidos de cada experimento proporcionaron información valiosa sobre la usabilidad del sistema y permitieron identificar oportunidades de mejora para las interfaces desarrolladas. Se observó que algunos usuarios expresaron sugerencias para optimizar la experiencia del usuario, como la implementación de ajustes para el tamaño de fuentes, un sistema de cambios más intuitivo mediante el uso de modales y la incorporación de una barra de búsqueda para facilitar la ubicación de artículos. Además, las reuniones semanales con el profesor guía y compañeros de carrera enriquecieron el proceso de iteración continua, lo que contribuyó a la evolución y refinamiento del sistema.

Aunque la evaluación se realizó en un restaurante de comida china, se reconoce la importancia de extender la prueba a otros tipos de establecimientos gastronómicos y negocios de mayor envergadura para evaluar la aplicabilidad y adaptabilidad del sistema en diversos entornos.

Los objetivos de la memoria están cumplidos de forma parcial. Se logró generar un modelo de solución para el problema de optimización, con información recabada a partir de entrevistas con expertos, junto con la experiencia previa del memorista. Sin embargo, como todos los expertos participantes, han trabajado solo en restaurantes de comida china, puede que se generó un sesgo en la solución a la que se llegó.

También se logró diseñar e implementar un modelo de datos, además de un back-end que cumple con las funcionalidades que surgieron durante el proceso de desarrollo que permitiera el manejo correcto de la solución al problema de optimización.

Por otro lado, en vez implementar un sistema TPV ya existente, finalmente se desarrolló uno desde cero debido a la libertad de creación que esto conlleva, sacrificando unas semanas de desarrollo del resto del sistema.

En cuanto al cuarto objetivo específico, se desarrolló una interfaz de usuario que muestra de manera clara el orden y que permita a los usuarios indicar cuando un artículo o pedido estuviese listo de forma sencilla, sin embargo, no se probó que estas interfaces fueran intuitivas, pues sólo se probaron con 2 usuarios a los cuales se les dio una capacitación de cómo usarlas, después de dejarlos explorar las vistas por su cuenta por unos minutos, esto fue un fallo por parte del memorista, pues no se diseñó una prueba pertinente con más usuarios para verificar que las interfaces fueran intuitivas.

El haber construido una aplicación desde sus bases ayuda al memorista a generar diversos aprendizajes. El primero es haber adquirido una profundización en el conocimiento del desarrollo web, debido a las diversas funcionalidades que se tuvieron que implementar siguiendo métodos que el memorista nunca había utilizado, como por ejemplo emplear Ajax para realizar cambios en tiempo real o “IRT” (por sus siglas en inglés *In Real Time*) y emplear *websockets* para activar cambios únicamente cuando fuera necesario. Otro aprendizaje es el haber podido plasmar los conocimientos en un rubro con el que creció el memorista, que en el transcurso del desarrollo de la memoria, este conocimiento se profundizó aún más. Y por último, a que la próxima vez que se quiera desarrollar una aplicación que requiera una gran interacción con usuarios, se diseñen las vistas con mayor detalle, pues durante el desarrollo de estas, se tuvieron que modificar muchas características, proceso que llevó más tiempo de lo esperado, tiempo que se pudo dedicar a realizar más pruebas del sistema.

Por otro lado, el prototipo no se considera terminado, quedan aspectos por mejorar, principalmente en cuanto a la estética de las vistas, desarrollar una mejor forma de añadir las prioridades (actualmente se añaden mediante las interfaces de usuario de manejo de base de datos que ofrece *Django*) y crear una vista para poder acceder a las mesas de forma rápida (actualmente se accede mediante una vista que muestra todos los pedidos del día).

Otros aspectos a mejorar son con respecto a la eficiencia del código, pues probando el caso de que llegan 20 pedidos de Menú para 10 (el artículo con más combinaciones), al sistema le toma alrededor de 10 segundos en procesar un cambio al interactuar con la vista de pedidos, al tomarle tanto tiempo al programa, puede generar errores de consistencia o inducir a errores en el funcionamiento del sistema. Al hablar con un experto con más de 20 años de experiencia en el rubro, nunca le ha sucedido este caso, sin embargo, no sería un caso tan raro para restaurantes de comida rápida, debido a su gran demanda, quizás no tengan un artículo de su carta con la misma cantidad que la de un Menú para 10 (11 artículos distintos), pero perfectamente podría llegar el caso de tener más de 30 pedidos; por ello, es necesario resolver este defecto.

Al tener en cuenta los comentarios de los usuarios del experimento para la vista TPV, cada uno podría ser perfectamente válido como funcionalidad a implementar para mejorar la experiencia del usuario. Específicamente, se sugiere añadir un ajuste para el tamaño de las fuentes, modificar el sistema de cambios por un modal en vez de obligar al usuario a interactuar de nuevo con las familias, reemplazar el scroll horizontal por un sistema de botones para desplazarse por las familias y artículos, e implementar una barra de búsqueda para un artículo. Adicionalmente, durante las reuniones semanales con el profesor guía y compañeros de carrera, se propuso integrar imágenes para identificar los artículos además del texto existente, esto para ayudar de manera visual al usuario que interactúe con el sistema.

Una vez terminado el proceso, hablando con conocidos del memorista y presentándoles la aplicación, se llegó a diferentes ideas de funcionalidades que podría tener el sistema, una de ellas es tener por defecto una cierta lista de artículos independiente de sus familias, para ser mostradas en vez mostrar un mensaje por defecto de “Seleccione una familia para empezar”, donde estos artículos serían los que se suelen pedir con mayor frecuencia, para así agilizar la toma de pedidos. La segunda idea fue la de habilitar un filtro para pedidos pendientes por los tipos de pedidos, esto es, mostrar los pedidos para servir separado de los pedidos para

llevar, para facilitar la lectura de los pedidos dentro de la cocina y poder identificarlos de manera más rápida.

Otra idea que surgió de estas “entrevistas” fue que se pudieran habilitar un sistema de sucursales para negocios con una mayor escala, esto se podría lograr fácilmente agregando una tabla con las sucursales y asociar las ventas, estaciones y colas a cada sucursal mediante un ID, adicionalmente, la persona que propuso esta idea, también propuso que se generara una especie de *dashboard* con la información más relevante como que platillo es el más pedido, conque se suele pedir un artículo, entre otros posibles datos que permitan mejorar el negocio.

En retrospectiva, la operatividad de los restaurantes que emplean mesas podría ser optimizada. Se podría aprovechar la existente interconexión wifi entre los dispositivos para adaptar la interfaz TPV a dispositivos más compactos, como los teléfonos inteligentes. Esta adaptación permitiría a los camareros registrar los pedidos directamente en el sistema, evitando así el proceso actual que requiere registrar primero los pedidos en papel y luego introducirlos al sistema mediante un dispositivo ubicado en la caja. Este análisis se basa en observaciones realizadas en un restaurante específico.

En conclusión, la aplicación TPV desarrollada representa una solución efectiva para mejorar la gestión y eficiencia de un restaurante de comida china. La priorización de preparación basada en colas ha demostrado ser una herramienta funcional para tomar decisiones informadas y agilizar el proceso de servicio. Asimismo, el sistema proporciona una base sólida para futuros desarrollos en el dominio, abriendo oportunidades para su aplicación en otros restaurantes chinos o establecimientos de comida rápida que busquen optimizar sus operaciones. Con las funcionalidades adicionales propuestas, el sistema podría convertirse en una herramienta aún más completa y versátil para la gestión de negocios gastronómicos, consolidándose como una solución eficiente y escalable para mejorar la eficiencia de este tipo de negocios.

Bibliografía

- [1] Aalen, Odd O.: *A linear regression model for the analysis of life times*. *Statistics in Medicine*, 8(8):907–925, 1989. <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.4780080803>, visitado el 23/06/2023.
- [2] Kenny, Vincent, Matthew Nathal y Spencer Saldana: *Heuristic algorithms*. Visited on Oct, 20:2018, 2014. https://optimization.mccormick.northwestern.edu/index.php/Heuristic_algorithms, visitado el 9/12/2022.
- [3] Pieskä, Sakari, Markus Liuska, Juhana Jauhiainen, Antti Auno y Delitaz Oy: *Intelligent restaurant system Smartmenu*. En *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, páginas 625–630, 2013. <https://ieeexplore.ieee.org/abstract/document/6719177>, visitado el 23/06/2023.
- [4] Wixon, Dennis y Chauncey Wilson: *Chapter 27 - The Usability Engineering Framework for Product Design and Evaluation*. En Helander, Marting G., Thomas K. Landauer y Prasad V. Prabhu (editores): *Handbook of Human-Computer Interaction (Second Edition)*, páginas 653–688. North-Holland, Amsterdam, second edition edición, 1997, ISBN 978-0-444-81862-1. <https://www.sciencedirect.com/science/article/pii/B9780444818621500935>, visitado el 27/06/2023.

Anexos

Anexo A

Instrumentos

A.1. Experimento 1

The image shows a digital survey form titled "Encuesta durante Intervención". At the top, it indicates "Not shared" and includes a red asterisk legend: "* Indicates required question". The form consists of several sections:

- ID ***: A text input field with the placeholder "Your answer".
- Número de errores ***: A text input field with the placeholder "Your answer".
- Número de ayudas ***: A text input field with the placeholder "Your answer".
- Tarea**: A radio button selection list with seven options: Tarea 1, Tarea 2, Tarea 3, Tarea 4, Tarea 5, Tarea 6, and Tarea 7.
- Errores**: A text input field with the placeholder "Your answer".

Figura A.1: Formulario durante intervención experimento 1.

Encuesta Post-Intervención

Not shared

* Indicates required question

ID *

Your answer _____

Edad *

Your answer _____

¿Qué tan complicado encontraste el flujo? *

Fácil 1 2 3 4 5 Dificil

○ ○ ○ ○ ○

¿Qué tareas te parecieron difíciles?

- Tarea 1
- Tarea 2
- Tarea 3
- Tarea 4
- Tarea 5
- Tarea 6
- Tarea 7

Cosas que te parecieron buenas

Your answer _____

Cosas que cambiarías

Your answer _____

Figura A.2: Encuesta post-intervención experimento 1.

A.2. Experimento 2

Reusmen

Prueba 1	0	0.00%
Prueba 2	0	0.00%
Prueba 3	0	0.00%
Prueba 4	0	0.00%
Prueba 5	0	0.00%
Prueba 6	0	0.00%
Prueba 7	0	0.00%
Prueba 8	0	0.00%
Prueba 9	0	0.00%

Figura A.3: Tabla que resume los resultados de las pruebas del experimento 2.

Combinaciones

Articulo	Cantidad	Sub
Menu x 2A	1	Carne Mongoliana
Menu x 2A	1	Chapsui de Pollo
Menu x 2A	2	Arroz Chaufan
Menu x 2A	1	Wantan

Articulos

Nombre	Espacio Utilizado	Tipo estacion	Familia
Carne Mongoliana	1	1	Vacunos
Chapsui de Pollo	1	1	Verduras
Wantan	2	2	Aperitivos
Arroz Chaufan	1	3	Arroz
Arrollado Primavera	1	2	Aperitivos

Prioridades

Familia	Tipo pedido	Prioridad
Aperitivos	Servir	1

Figura A.4: Configuración experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pedidos

Id	Orden	Tipo
	1	Llevar
	2	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	3	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	1	Carne Mongoliana	Preparando	1 x Pedido 1
Platillos_01	3	Carne Mongoliana	Encolado	3 x Pedido 2

Pruebas:

- 1
- 2
- 3
- 4
- 5

Figura A.5: Prueba 1 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pedidos

Id	Orden	Tipo
	1	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	15	1

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	6	Carne Mongoliana	Preparando	6 x Pedido 1
Platillos_01	6	Carne Mongoliana	Encolado	6 x Pedido 1
Platillos_01	3	Carne Mongoliana	Encolado	3 x Pedido 1

Pruebas:

- 1
- 2
- 3
- 4
- 5

Figura A.6: Prueba 2 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Frituras_01	2	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
	1	1 Llevar
	2	2 Servir

Artículos

Nombre	Cantidad	Pedido
Wantan	1	1
Arrollado Primavera	1	1
Wantan	1	2

Resultado

Estación	Cantidad	Artículo	Estado	Origen
Frituras_01	1	Wantan / Arrollado Primavera	Preparando	1 x Pedido 1
Frituras_01	1	Wantan	Encolado	1 x Pedido 2
Frituras_01	1	Arrollado Primavera / Wantan	Encolado	1 x Pedido 1

Figura A.7: Prueba 3 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6
Frituras_01	2	6
Arroz_01	3	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar

Artículos

Nombre	Cantidad	Pedido
Menú x 2A	1	1

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	1	Wantan	Preparando	1 x Pedido 1
Arroz_01	2	Arroz Chaufan	Preparando	2 x Pedido 1
Platillos_01	1	Carne Mongoliana / Chapsui de Pollo	Preparando	1 x Pedido 1
Platillos_01	1	Chapsui de Pollo / Carne Mongoliana	Encolado	1 x Pedido 1

Figura A.8: Prueba 4 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6
Platillos_02	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar
2	2	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	1	2
Chapsui de Pollo	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01 / Platillos_02	1	Carne Mongoliana	Preparando	1 x Pedido 1
Platillos_02 / Platillos_01	1	Carne Mongoliana / Chapsui de Pollo	Preparando	1 x Pedido 2
Platillos_01 o Platillos_02	1	Chapsui de Pollo / Carne Mongoliana	Encolado	1 x Pedido 2

Figura A.9: Prueba 5 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Frituras_01	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar

Artículos

Nombre	Cantidad	Pedido
Wantan	15	1

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	3	Wantan	Preparando	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1

Figura A.10: Prueba 6 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Fecha actual: 01/01/2023 13:00

Pedidos

Id	Orden	Tipo	Programado
1	1	Llevar	02/01/2023 13:00
2	2	Llevar	-

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	1	Carne Mongoliana	Preparando	1 x Pedido 2

Figura A.11: Prueba 7 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar
2	2	Llevar
3	3	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	3	2
Carne Mongoliana	4	3

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	1	Carne Mongoliana	Preparando	1 x Pedido 1
Platillos_01	6	Carne Mongoliana	Encolado	3 x Pedido 2 3 x Pedido 3
Platillos_01	1	Carne Mongoliana	Encolado	1 x Pedido 3

Figura A.12: Prueba 8 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6
Frituras_01	2	6
Arroz_01	3	6
Platillos_02	1	6
Frituras_02	2	6
Arroz_02	3	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
	1	Llevar
	2	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Menu x 2A	1	1
Chapsui Pollo	1	2
Menu x 2A	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	1	Wantan	Preparando	1 x Pedido 1
Frituras_02	1	Wantan	Preparando	1 x Pedido 2
Arroz_01	2	Arroz Chaufan	Preparando	2 x Pedido 1
Arroz_02	2	Arroz Chaufan	Preparando	2 x Pedido 2
Platillos_01 / Platillos_02	2	Carne Mongoliana	Preparando	2 x Pedido 1
Platillos_02 / Platillos_01	1	Chapsui de Pollo	Preparando	1 x Pedido 1
Platillos_01 / Platillos_02	1	Carne Mongoliana	Encolado	1 x Pedido 2
Platillos_02 / Platillos_01	2	Chapsui de Pollo	Encolado	2 x Pedido 2

Figura A.13: Prueba 9 experimento 2.

Anexo B

Resultados

B.1. Experimento 1

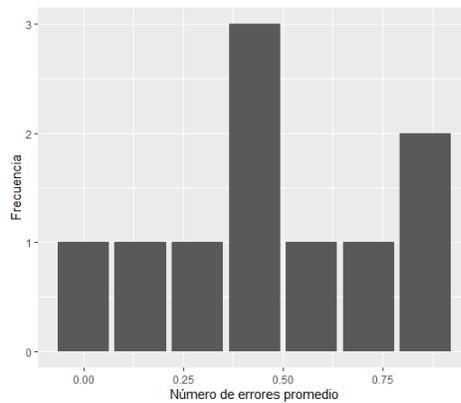


Figura B.1: Gráfico de barras de número de errores promedio del experimento 1.

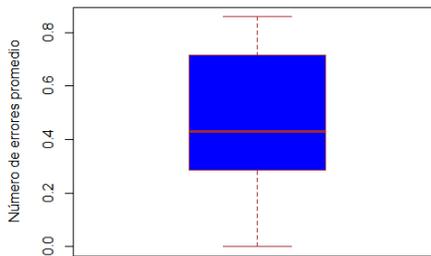


Figura B.2: Diagrama de caja de número de errores promedio del experimento 1.

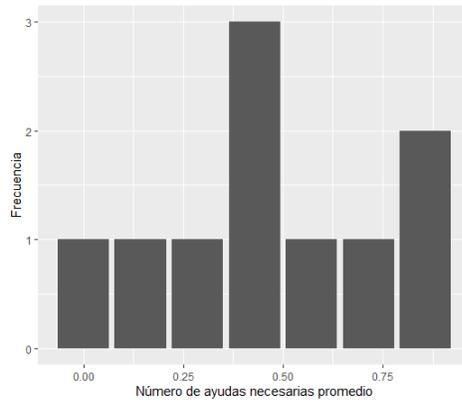


Figura B.3: Gráfico de barras de número de ayudas promedio del experimento 1.

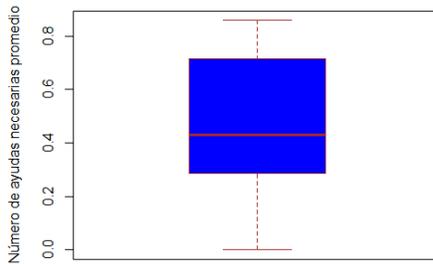


Figura B.4: Diagrama de caja de número de ayudas promedio del experimento 1.

B.2. Experimento 2

Reusmen		
Prueba 1	5	100.00%
Prueba 2	5	100.00%
Prueba 3	5	100.00%
Prueba 4	5	100.00%
Prueba 5	5	100.00%
Prueba 6	5	100.00%
Prueba 7	5	100.00%
Prueba 8	5	100.00%
Prueba 9	5	100.00%

Figura B.7: Tabla de resultados de las pruebas del experimento 2.

ID	Tarea	# de errores	# de ayudas	Completado	Errores	ID	Tarea	# de errores	# de ayudas	Completado	Errores
1	1	0	0	1	-	5	1	0	0	1	-
1	2	0	0	1	-	5	2	0	0	1	-
1	3	0	0	1	-	5	3	0	0	1	-
1	4	1	1	1	No encontraba el botón de notas.	5	4	0	0	1	-
1	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.	5	5	0	0	1	-
1	6	0	0	1	No sabía que se tenía que seleccionar el nuevo artículo en la carta.	5	6	0	0	1	-
1	7	0	0	1	-	5	7	0	0	1	-
2	1	0	0	1	-	6	1	0	0	1	-
2	2	0	0	1	-	6	2	0	0	1	-
2	3	0	0	1	-	6	3	0	0	1	-
2	4	1	1	1	No encontraba el botón de notas.	6	4	1	1	1	No encontraba el botón de notas.
2	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.	6	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.
2	6	0	0	1	No sabía que se tenía que seleccionar el nuevo artículo en la carta.	6	6	0	0	1	-
2	7	0	0	1	-	6	7	0	0	1	-
3	1	0	0	1	-	7	1	0	0	1	-
3	2	1	1	1	Presiona botón de "Eliminar" todo el pedido en vez de un artículo	7	2	0	0	1	-
3	3	0	0	1	-	7	3	0	0	1	-
3	4	1	1	1	No encontraba el botón de notas.	7	4	1	1	1	No encontraba el botón de notas.
3	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.	7	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.
3	6	1	1	1	No encontraba el botón de Nuevo pedido.	7	6	0	0	1	-
4	1	0	0	1	-	8	1	0	0	1	-
4	2	0	0	1	-	8	2	0	0	1	-
4	3	0	0	1	-	8	3	0	0	1	-
4	4	0	0	1	-	8	4	1	1	1	No encontraba el botón de notas.
4	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.	8	5	2	2	1	No sabía que para cambiar un artículo había que abrir las notas.
4	6	0	0	1	No sabía que se tenía que seleccionar el nuevo artículo en la carta.	8	6	0	0	1	-
4	7	0	0	1	-	8	7	0	0	1	-

Figura B.5: Resultados en crudo de los datos recopilados durante la experiencia del experimento 1.

ID	Edad	¿Qué tan complicado encontraste el flujo?	¿Qué tareas se parecieron difíciles?	Cosas que se parecieron buenas	Cosas que cambiarías
1	23	2 5			Que se pudiera buscar un artículo por su nombre Que se agregaran más botones para desplazarse Que aparezca un mini menú al hacer un cambio en vez de seleccionarlo desde el menú original
2	20	3 4 5			Dibujo de notas
3	52	4 4 5 6 7			Tamaño de la letra
4	26	1 5	Poder agregar datos de cliente		Botones para moverse en el menú Al hacer un cambio, que aparezca algo para seleccionar el artículo por el que se quiera cambiar
5	24	1			
6	23	1 5			
7	23	2 4 5			Cambiar los iconos de las notas
8	20	2 4 5			
9	47	4 4 5 6			
10	57	4 3 4 5 6			Agrandar las letras

Figura B.6: Resultados en crudo de los datos recopilados después de la experiencia del experimento 1.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pedidos

Id	Orden	Tipo
1	1	Llevar
2	2	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	3	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	1	Carne Mongoliana	Preparando	1 x Pedido 1
Platillos_01	3	Carne Mongoliana	Encolado	3 x Pedido 2

Pruebas:

- 1
- 2
- 3
- 4
- 5

Figura B.8: Resultados de la prueba 1 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pedidos

Id	Orden	Tipo
1	1	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	15	1

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	6	Carne Mongoliana	Preparando	6 x Pedido 1
Platillos_01	6	Carne Mongoliana	Encolado	6 x Pedido 1
Platillos_01	3	Carne Mongoliana	Encolado	3 x Pedido 1

Pruebas:

- 1
- 2
- 3
- 4
- 5

Figura B.9: Resultados de la prueba 2 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Frituras_01	2	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
	1	Llevar
	2	Servir

Artículos

Nombre	Cantidad	Pedido
Wantan	1	1
Arrollado Primavera	1	1
Wantan	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	1	Wantan / Arrollado Primavera	Preparando	1 x Pedido 1
Frituras_01	1	Wantan	Encolado	1 x Pedido 2
Frituras_01	1	Arrollado Primavera / Wantan	Encolado	1 x Pedido 1

Figura B.10: Resultados de la prueba 3 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6
Frituras_01	2	6
Arroz_01	3	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar

Artículos

Nombre	Cantidad	Pedido
Menú x 2A	1	1

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	1	Wantan	Preparando	1 x Pedido 1
Arroz_01	2	Arroz Chaufan	Preparando	2 x Pedido 1
Platillos_01	1	Carne Mongoliana / Chapsui de Pollo	Preparando	1 x Pedido 1
Platillos_01	1	Chapsui de Pollo / Carne Mongoliana	Encolado	1 x Pedido 1

Figura B.11: Resultados de la prueba 4 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6
Platillos_02	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar
2	2	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	1	2
Chapsui de Pollo	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01 / Platillos_02	1	Carne Mongoliana	Preparando	1 x Pedido 1
Platillos_02 / Platillos_01	1	Carne Mongoliana / Chapsui de Pollo	Preparando	1 x Pedido 2
Platillos_01 o Platillos_02	1	Chapsui de Pollo / Carne Mongoliana	Encolado	1 x Pedido 2

Figura B.12: Resultados de la prueba 5 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Frituras_01	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar

Artículos

Nombre	Cantidad	Pedido
Wantan	15	1

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	3	Wantan	Preparando	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1
Frituras_01	3	Wantan	Encolado	3 x Pedido 1

Figura B.13: Resultados de la prueba 6 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Fecha actual: 01/01/2023 13:00

Pedidos

Id	Orden	Tipo	Programado
1	1	Llevar	02/01/2023 13:00
2	2	Llevar	-

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	1	Carne Mongoliana	Preparando	1 x Pedido 2

Figura B.14: Resultados de la prueba 7 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
	1	1 Llevar
	2	2 Llevar
	3	3 Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Carne Mongoliana	3	2
Carne Mongoliana	4	3

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Platillos_01	1	Carne Mongoliana	Preparando	1 x Pedido 1
				3 x Pedido 2
Platillos_01	6	Carne Mongoliana	Encolado	3 x Pedido 3
Platillos_01	1	Carne Mongoliana	Encolado	1 x Pedido 3

Figura B.15: Resultados de la prueba 8 experimento 2.

Estaciones

Nombre	Tipo	Capacidad
Platillos_01	1	6
Frituras_01	2	6
Arroz_01	3	6
Platillos_02	1	6
Frituras_02	2	6
Arroz_02	3	6

Pruebas:

- 1
- 2
- 3
- 4
- 5

Pedidos

Id	Orden	Tipo
1	1	Llevar
2	2	Llevar

Artículos

Nombre	Cantidad	Pedido
Carne Mongoliana	1	1
Menu x 2A	1	1
Chapsui Pollo	1	2
Menu x 2A	1	2

Resultado

Estacion	Cantidad	Articulo	Estado	Origen
Frituras_01	1	Wantan	Preparando	1 x Pedido 1
Frituras_02	1	Wantan	Preparando	1 x Pedido 2
Arroz_01	2	Arroz Chaufan	Preparando	2 x Pedido 1
Arroz_02	2	Arroz Chaufan	Preparando	2 x Pedido 2
Platillos_01 / Platillos_02	2	Carne Mongoliana	Preparando	2 x Pedido 1
Platillos_02 / Platillos_01	1	Chapsui de Pollo	Preparando	1 x Pedido 1
Platillos_01 / Platillos_02	1	Carne Mongoliana	Encolado	1 x Pedido 2
Platillos_02 / Platillos_01	2	Chapsui de Pollo	Encolado	2 x Pedido 2

Figura B.16: Resultados de la prueba 9 experimento 2.

B.3. Experimento 3

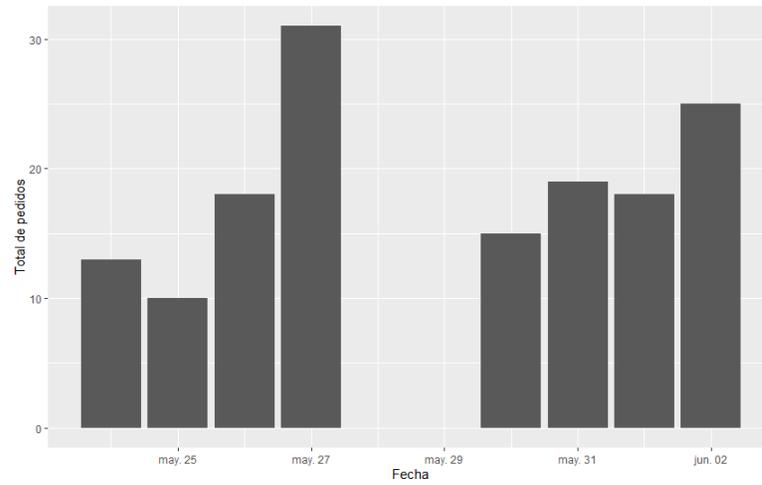


Figura B.17: Gráfico de barras de número de pedidos durante la prueba con sistema del experimento 3.

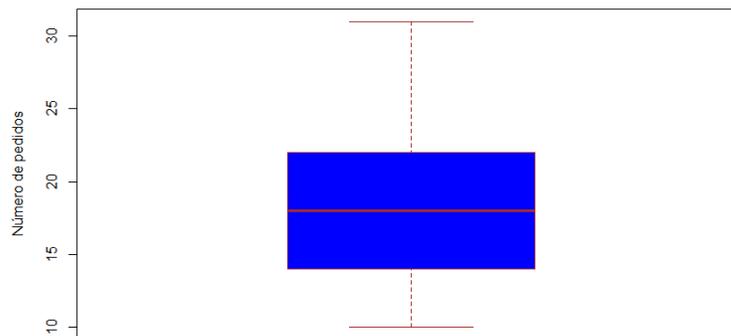


Figura B.18: Diagrama de caja de número de pedidos durante la prueba con sistema del experimento 3.

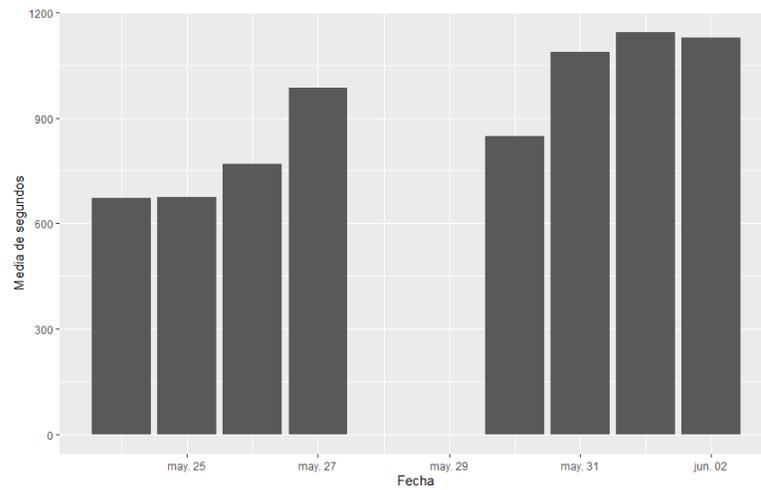


Figura B.19: Gráfico de barras de media de tiempo en segundo que demora un pedido durante la prueba con sistema del experimento 3.

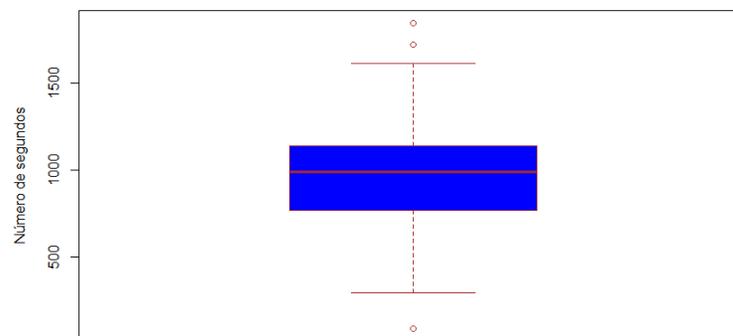


Figura B.20: Diagrama de caja de tiempo en segundo que demora un pedido durante la prueba con sistema del experimento 3.

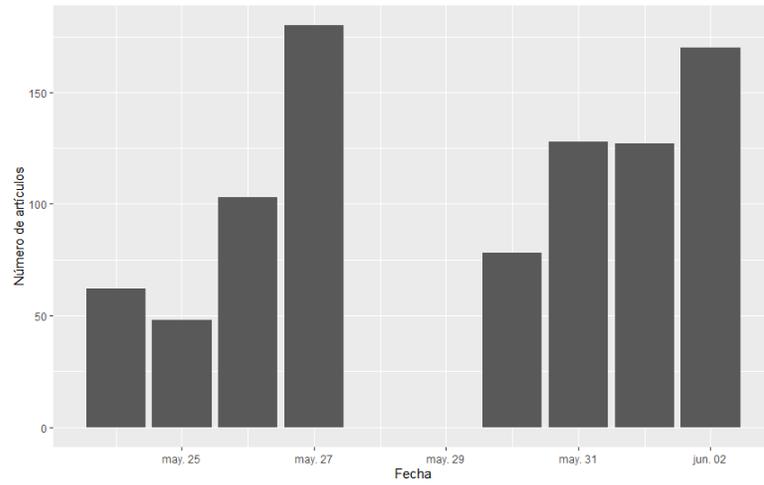


Figura B.21: Gráfico de barras de total de artículos por día durante la prueba con sistema del experimento 3.

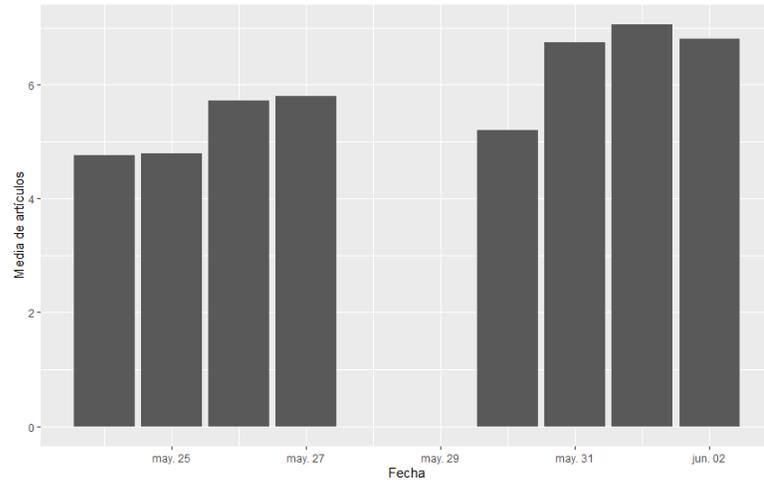


Figura B.22: Gráfico de barras de media de artículos de un pedido por día durante la prueba con sistema del experimento 3.

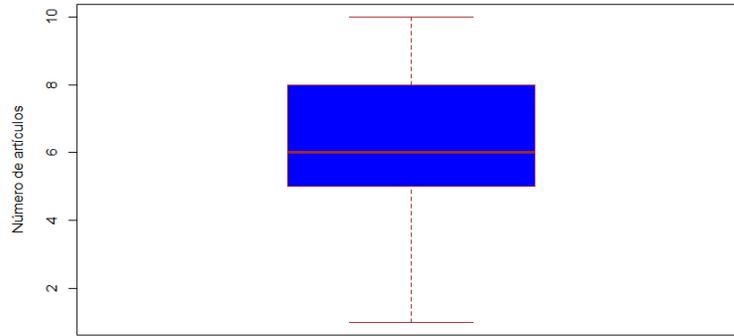


Figura B.23: Diagrama de caja de número de artículos durante la prueba con sistema del experimento 3.

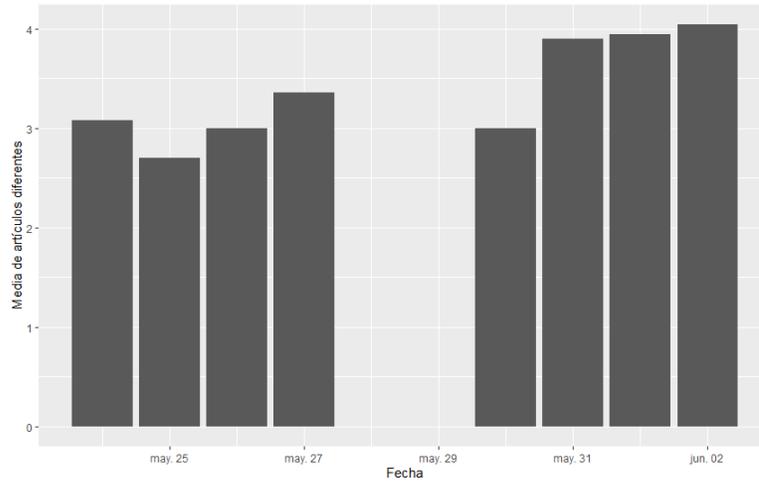


Figura B.24: Gráfico de barras de media de artículos distintos de un pedido durante la prueba con sistema del experimento 3.

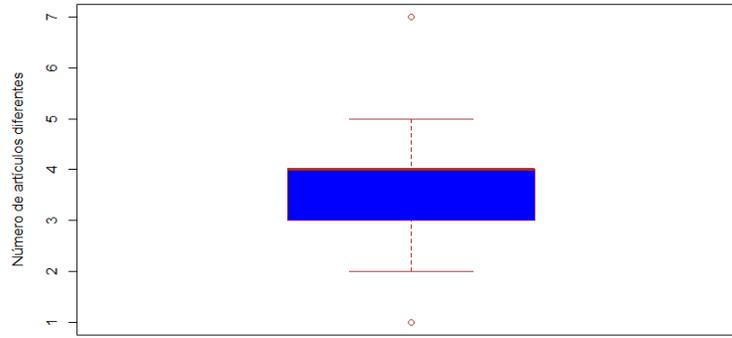


Figura B.25: Diagrama de caja de media de artículos distintos de un pedido durante la prueba con sistema del experimento 3.

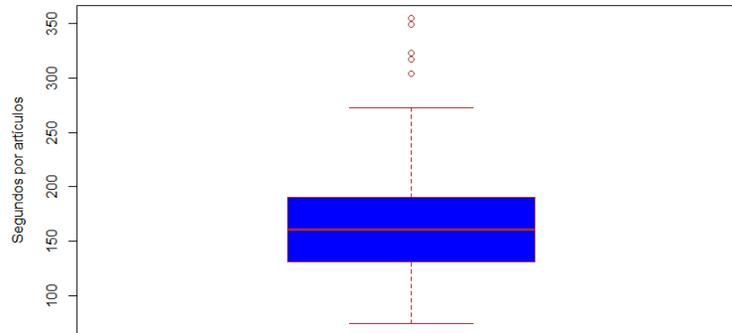


Figura B.26: Diagrama de caja de tiempo en segundos sobre la cantidad de artículos de un pedido durante la prueba con sistema del experimento 3.

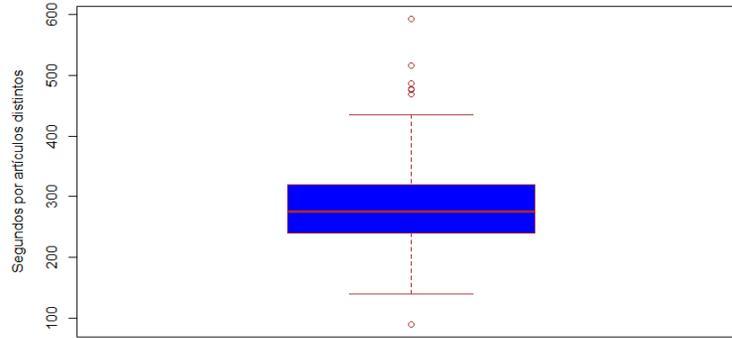


Figura B.27: Diagrama de caja de tiempo en segundos sobre la cantidad de artículos distintos de un pedido durante la prueba con sistema del experimento 3.

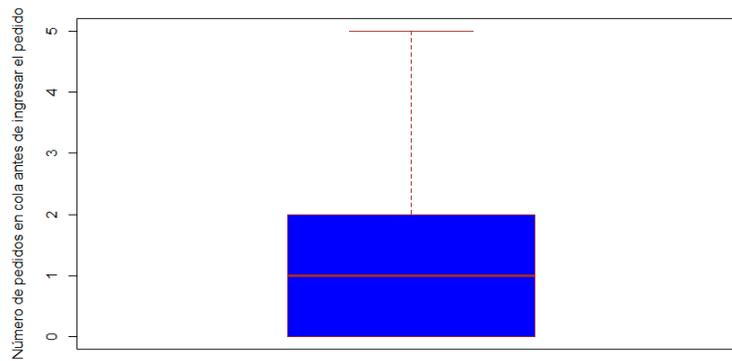


Figura B.28: Diagrama de caja del número de pedidos previamente en cola a la hora del ingreso de un nuevo pedido durante la prueba con sistema del experimento 3.

Tipo	Error	Frecuencia
Sistema	Tabletas se bloqueaban por inactividad	32
	Color de artículo equivocado	20
	El artículo solicitado no se encuentra en el sistema	18
	Muestra cantidad errónea de cantidad a preparar	10
	Aparece botón "Listo" de un pedido listo multiple veces	1
Usuario	Se ingresó la cantidad de un pedido de forma equivocada en la vista dentro de la cocina	10
	Se ingresó de forma errónea el pedido desde la vista TPV	2

Figura B.29: Errores ocurridos durante la prueba con sistema del experimento 3.

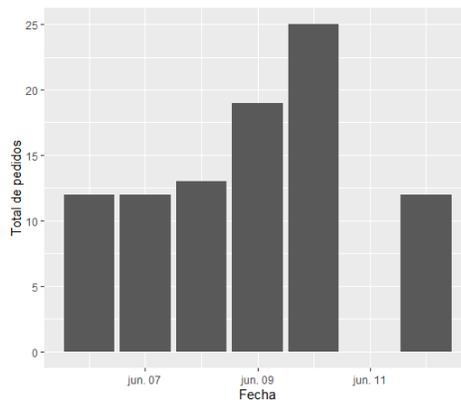


Figura B.38: Gráfico de barras de número de pedidos durante la prueba sin sistema del experimento 3.

Pedido	Programado	Mesa	Día ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
1	<input type="checkbox"/>		24/05/2023	12:45:34	12:55:23	0:09:49	589	4	4	2	0	147.25
2	<input type="checkbox"/>	1	24/05/2023	13:04:17	13:24:30	0:20:13	1,213	10	7	7	0	121.30
3	<input type="checkbox"/>		24/05/2023	13:16:46	13:26:04	0:09:18	558	6	4	4	1	93.00
4	<input type="checkbox"/>	9	24/05/2023	13:55:10	14:10:39	0:15:29	929	5	5	4	0	185.80
5	<input type="checkbox"/>		24/05/2023	14:24:14	14:38:16	0:14:02	842	6	6	3	0	140.33
6	<input type="checkbox"/>	15	24/05/2023	14:39:12	14:48:58	0:09:46	586	5	3	3	0	117.20
7	<input type="checkbox"/>		24/05/2023	14:47:50	14:54:28	0:06:38	398	2	2	2	1	199.00
8	<input type="checkbox"/>		24/05/2023	14:50:03	15:01:30	0:11:27	687	5	3	1	1	137.40
9	<input type="checkbox"/>		24/05/2023	15:13:20	15:19:15	0:05:55	355	1	1	0	0	355.00
10	<input type="checkbox"/>		24/05/2023	15:20:10	15:34:51	0:14:41	881	5	5	4	0	176.20
11	<input type="checkbox"/>		24/05/2023	16:43:51	16:45:20	0:01:29	89	1	1	1	0	89.00
12	<input type="checkbox"/>		24/05/2023	20:10:14	20:15:34	0:05:20	320	4	4	1	0	80.00
13	<input type="checkbox"/>		24/05/2023	20:30:25	20:51:41	0:21:16	1,276	8	8	5	0	159.50

Figura B.30: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 24/05/2023 del experimento 3.

Pedido	Programado	Mesa	Día ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
14	<input type="checkbox"/>		25/05/2023	12:30:51	12:40:05	0:09:14	554	6	6	4	0	92.33
15	<input type="checkbox"/>		25/05/2023	13:42:02	13:49:38	0:07:36	456	3	3	2	0	152.00
16	<input type="checkbox"/>	13	25/05/2023	13:46:12	13:58:52	0:12:40	760	5	5	3	1	152.00
17	<input type="checkbox"/>	15	25/05/2023	13:57:51	14:15:54	0:18:03	1,083	6	6	4	1	180.50
18	<input type="checkbox"/>	1	25/05/2023	14:34:12	14:45:15	0:11:03	663	4	4	2	0	165.75
19	<input type="checkbox"/>		25/05/2023	14:39:47	14:57:08	0:17:21	1,041	6	6	4	1	173.50
20	<input type="checkbox"/>		25/05/2023	15:40:23	15:46:31	0:06:08	368	2	2	1	0	184.00
21	<input type="checkbox"/>		25/05/2023	16:04:56	16:11:08	0:06:12	372	2	2	1	0	186.00
22	<input type="checkbox"/>		25/05/2023	19:45:43	20:04:56	0:19:13	1,153	10	10	5	0	115.30
23	<input type="checkbox"/>		25/05/2023	20:48:56	20:53:54	0:04:58	298	4	4	1	0	74.50
												298.00

Figura B.31: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 25/05/2023 del experimento 3.

Pedido	Programado	Mesa	Día ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Diff/Art	Diff/Art dif
24	<input type="checkbox"/>		26/05/2023	12:25:14	12:42:54	0:17:40	1,060	8	8	4	0	132.50
25	<input type="checkbox"/>		26/05/2023	13:16:08	13:31:45	0:15:37	937	8	8	2	0	117.13
26	<input type="checkbox"/>		26/05/2023	13:19:13	13:35:10	0:15:57	957	10	10	2	1	95.70
27	<input type="checkbox"/>		26/05/2023	13:25:41	13:31:48	0:06:07	367	2	2	1	2	183.50
28	<input type="checkbox"/>		26/05/2023	13:42:10	13:50:04	0:07:54	474	3	3	2	0	158.00
29	<input type="checkbox"/>		26/05/2023	13:49:17	13:59:42	0:10:25	625	4	4	2	1	156.25
30	<input type="checkbox"/>	1	26/05/2023	14:05:48	14:24:51	0:19:03	1,143	8	8	3	0	142.88
31	<input type="checkbox"/>		26/05/2023	14:16:15	14:21:09	0:04:54	294	3	3	1	1	98.00
32	<input type="checkbox"/>		26/05/2023	14:26:08	14:44:54	0:16:46	1,006	8	8	5	0	125.75
33	<input type="checkbox"/>		26/05/2023	14:34:49	14:43:14	0:08:25	505	4	4	3	1	126.25
34	<input type="checkbox"/>		26/05/2023	14:49:11	14:57:41	0:08:30	510	4	4	3	0	127.50
35	<input type="checkbox"/>		26/05/2023	15:13:42	15:35:23	0:21:41	1,301	8	8	5	0	162.62
36	<input type="checkbox"/>		26/05/2023	15:49:16	16:03:40	0:14:24	864	5	5	4	0	172.80
37	<input type="checkbox"/>		26/05/2023	15:57:51	16:16:15	0:18:24	1,104	8	8	5	1	138.00
38	<input type="checkbox"/>		26/05/2023	16:48:19	16:54:08	0:05:49	349	1	1	1	0	349.00
39	<input type="checkbox"/>		26/05/2023	19:10:48	19:28:01	0:17:13	1,033	8	8	5	0	129.12
40	<input type="checkbox"/>		26/05/2023	19:26:14	19:31:08	0:04:54	294	3	3	1	1	98.00
41	<input type="checkbox"/>	12	26/05/2023	20:41:16	20:58:14	0:16:58	1,018	8	8	5	0	127.25
												203.60

Figura B.32: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 26/05/2023 del experimento 3.

Pedido	Programado	Mesa	Día ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Diff/Art	Diff/Art dif
42			27/05/2023	12:41:08	12:56:46	0:15:38	938	5	4	0	187.60	234.50
43			27/05/2023	13:36:49	13:46:38	0:09:49	589	5	3	0	117.80	196.33
44			27/05/2023	13:38:42	13:52:07	0:13:25	805	4	2	1	201.25	402.50
45		1	27/05/2023	13:39:46	13:58:18	0:18:32	1,112	9	3	2	123.56	370.67
46			27/05/2023	13:45:35	13:57:16	0:11:41	701	3	2	3	233.67	350.50
47			27/05/2023	13:48:31	13:59:05	0:10:34	634	2	2	3	317.00	317.00
48			27/05/2023	13:53:15	14:05:27	0:12:12	732	3	3	3	244.00	244.00
49			27/05/2023	14:04:46	14:13:51	0:09:05	545	5	3	1	109.00	181.67
50		4	27/05/2023	14:07:58	14:21:54	0:13:56	836	8	3	1	104.50	278.67
51		15	27/05/2023	14:09:06	14:30:42	0:21:36	1,296	7	4	2	185.14	324.00
52			27/05/2023	14:18:53	14:31:54	0:13:01	781	4	2	2	195.25	390.50
53			27/05/2023	14:26:58	14:39:45	0:12:47	767	3	2	2	255.67	383.50
54		19	27/05/2023	14:28:41	14:48:55	0:20:14	1,214	8	4	3	151.75	303.50
55			27/05/2023	14:31:34	14:46:20	0:14:46	886	4	3	3	221.50	295.33
56			27/05/2023	14:32:57	14:49:16	0:16:19	979	7	3	3	139.86	326.33
57			27/05/2023	14:36:47	15:03:42	0:26:55	1,615	5	4	4	323.00	403.75
58		11	27/05/2023	14:39:08	15:04:33	0:25:25	1,525	10	5	5	152.50	305.00
59			27/05/2023	14:49:53	15:05:45	0:15:52	952	6	3	2	158.67	317.33
60			27/05/2023	14:54:38	15:06:34	0:11:56	716	4	3	3	179.00	238.67
61		8	27/05/2023	15:01:59	15:24:41	0:22:42	1,362	5	4	4	272.40	340.50
62			27/05/2023	15:06:10	15:31:46	0:25:36	1,536	10	5	2	153.60	307.20
63			27/05/2023	15:14:23	15:32:45	0:18:22	1,102	5	4	2	220.40	275.50
64		1	27/05/2023	15:17:05	15:37:10	0:20:05	1,205	10	5	3	120.50	241.00
65		15	27/05/2023	15:23:46	15:41:42	0:17:56	1,076	8	4	4	134.50	269.00
66			27/05/2023	15:28:15	15:48:00	0:19:45	1,185	5	4	4	237.00	296.25
67			27/05/2023	15:31:45	15:52:14	0:20:29	1,229	5	4	5	245.80	307.25
68			27/05/2023	15:37:53	15:49:08	0:11:15	675	5	3	3	135.00	225.00
69			27/05/2023	17:16:08	17:27:52	0:11:44	704	4	2	0	176.00	352.00
70			27/05/2023	19:39:20	19:52:48	0:13:28	808	6	2	0	134.67	404.00
71			27/05/2023	20:38:56	20:55:15	0:16:19	979	7	4	0	139.86	244.75
72			27/05/2023	20:43:43	21:01:53	0:18:10	1,090	8	5	1	136.25	218.00

Figura B.33: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 27/05/2023 del experimento 3.

Pedido	Programado	Mesa	Dia Ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
73	<input type="checkbox"/>		30/05/2023	12:51:27	13:05:59	0:14:32	872	5	4	0	174.40	218.00
74	<input type="checkbox"/>		30/05/2023	12:58:24	13:07:16	0:08:52	532	4	2	1	133.00	266.00
75	<input type="checkbox"/>		30/05/2023	12:59:51	13:09:58	0:10:07	607	2	2	2	303.50	303.50
76	<input type="checkbox"/>		30/05/2023	13:03:54	13:12:58	0:09:04	544	4	2	3	136.00	272.00
77	<input type="checkbox"/>		30/05/2023	13:06:48	13:19:51	0:13:03	783	5	3	3	156.60	261.00
78	<input type="checkbox"/>	15	30/05/2023	13:24:35	13:38:32	0:13:57	837	4	4	0	209.25	209.25
79	<input type="checkbox"/>		30/05/2023	13:29:46	13:43:08	0:13:22	802	4	2	1	200.50	401.00
80	<input type="checkbox"/>	1	30/05/2023	13:36:24	13:52:27	0:16:03	963	5	4	2	192.60	240.75
81	<input type="checkbox"/>		30/05/2023	13:48:57	14:01:13	0:12:16	736	3	3	1	245.33	245.33
82	<input type="checkbox"/>		30/05/2023	13:52:07	14:09:36	0:17:29	1049	6	3	2	174.83	349.67
83	<input type="checkbox"/>		30/05/2023	13:58:16	14:14:08	0:15:52	952	8	2	2	119.00	476.00
84	<input type="checkbox"/>		30/05/2023	14:34:43	14:51:56	0:17:13	1033	8	5	0	129.13	206.60
85	<input type="checkbox"/>		30/05/2023	14:38:51	14:58:38	0:19:47	1187	6	2	1	197.83	593.50
86	<input type="checkbox"/>		30/05/2023	15:41:48	15:58:46	0:16:58	1018	8	5	0	127.25	203.60
87	<input type="checkbox"/>		30/05/2023	19:48:18	20:01:53	0:13:35	815	6	2	0	135.83	407.50

Figura B.34: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 30/05/2023 del experimento 3.

Pedido	Programado	Mesa	Dia Ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
88	<input type="checkbox"/>		31/05/2023	12:31:10	12:48:26	0:17:16	1,036	5	5	4	207.20	259.00
89	<input type="checkbox"/>		31/05/2023	13:08:48	13:23:08	0:14:20	860	5	5	4	172.00	215.00
90	<input type="checkbox"/>		31/05/2023	13:12:56	13:26:43	0:13:47	827	4	4	2	206.75	413.50
91	<input type="checkbox"/>		31/05/2023	13:21:08	13:35:28	0:14:20	860	6	6	2	143.33	430.00
92	<input type="checkbox"/>		31/05/2023	13:26:18	13:42:54	0:16:36	996	5	5	4	199.20	249.00
93	<input type="checkbox"/>		31/05/2023	13:32:15	13:48:32	0:16:17	977	6	6	2	162.83	244.25
94	<input type="checkbox"/>		31/05/2023	13:39:08	13:57:58	0:18:50	1,130	7	7	2	161.43	282.50
95	<input type="checkbox"/>		31/05/2023	13:42:47	14:01:43	0:18:56	1,136	7	7	4	162.29	284.00
96	<input type="checkbox"/>	15	31/05/2023	13:46:24	14:06:41	0:18:17	1,097	5	5	3	219.40	274.25
97	<input type="checkbox"/>		31/05/2023	13:53:10	14:21:52	0:28:42	1,722	10	10	7	172.20	246.00
98	<input type="checkbox"/>	1	31/05/2023	13:59:53	14:16:51	0:16:58	1,018	5	5	3	203.60	254.50
99	<input type="checkbox"/>		31/05/2023	14:08:34	14:31:32	0:22:58	1,378	8	8	2	172.25	275.60
100	<input type="checkbox"/>		31/05/2023	14:15:32	14:37:15	0:21:43	1,303	7	7	3	186.14	434.33
101	<input type="checkbox"/>		31/05/2023	14:27:14	14:48:12	0:20:58	1,258	10	10	4	125.80	314.50
102	<input type="checkbox"/>		31/05/2023	14:38:45	14:57:08	0:18:23	1,103	8	8	5	137.87	220.60
103	<input type="checkbox"/>		31/05/2023	16:21:53	16:42:16	0:20:23	1,223	10	10	0	122.30	305.75
104	<input type="checkbox"/>		31/05/2023	19:23:42	19:32:12	0:08:30	510	4	4	3	127.50	170.00
105	<input type="checkbox"/>		31/05/2023	19:55:10	20:13:01	0:17:51	1,071	8	8	0	133.87	357.00
106	<input type="checkbox"/>	12	31/05/2023	20:34:29	20:53:41	0:19:12	1,152	8	8	0	144.00	288.00

Figura B.35: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 31/05/2023 del experimento 3.

Pedido	Programado	Mesa	Día ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
107	<input type="checkbox"/>		01/06/2023	12:23:45	12:41:15	0:17:30	1,050	8	8	3	0	131.25
108	<input type="checkbox"/>		01/06/2023	13:34:45	13:51:58	0:17:13	1,033	8	8	5	0	129.13
109	<input type="checkbox"/>		01/06/2023	13:42:13	13:59:42	0:17:29	1,049	6	6	4	1	174.83
110	<input type="checkbox"/>	15	01/06/2023	13:46:45	14:09:45	0:23:00	1,380	8	8	4	2	172.50
111	<input type="checkbox"/>	1	01/06/2023	13:51:14	14:15:35	0:24:21	1,461	10	10	3	3	146.10
112	<input type="checkbox"/>		01/06/2023	13:58:45	14:19:20	0:20:35	1,235	5	5	4	3	247.00
113	<input type="checkbox"/>	19	01/06/2023	14:05:54	14:24:06	0:18:12	1,092	5	5	4	3	218.40
114	<input type="checkbox"/>	9	01/06/2023	14:13:15	14:35:45	0:22:30	1,350	7	7	5	3	192.86
115	<input type="checkbox"/>		01/06/2023	14:20:38	14:34:53	0:14:15	855	5	5	4	2	171.00
116	<input type="checkbox"/>		01/06/2023	14:26:08	14:52:43	0:26:35	1,595	8	8	5	2	199.37
117	<input type="checkbox"/>		01/06/2023	14:35:56	14:58:14	0:22:18	1,338	10	10	4	1	133.80
118	<input type="checkbox"/>		01/06/2023	14:46:11	15:07:41	0:21:30	1,290	7	7	4	2	184.29
119	<input type="checkbox"/>		01/06/2023	15:03:19	15:24:51	0:21:32	1,292	8	8	5	1	161.50
120	<input type="checkbox"/>		01/06/2023	15:53:15	16:11:52	0:18:37	1,117	9	9	4	0	124.11
121	<input type="checkbox"/>		01/06/2023	19:45:10	20:01:45	0:16:35	995	6	6	4	0	165.83
122	<input type="checkbox"/>		01/06/2023	20:03:18	20:21:54	0:18:36	1,116	9	9	4	0	124.00
123	<input type="checkbox"/>		01/06/2023	20:16:48	20:23:16	0:06:28	388	3	3	1	1	129.33
124	<input type="checkbox"/>		01/06/2023	20:48:36	21:04:12	0:15:36	936	5	5	4	0	187.20
												350.00
												206.60
												262.25
												345.00
												487.00
												308.75
												273.00
												270.00
												213.75
												319.00
												334.50
												322.50
												258.40
												279.25
												248.75
												279.00
												388.00
												234.00

Figura B.36: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 01/06/2023 del experimento 3.

Pedido	Programado	Mesa	Día Ingreso Y	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
125	<input type="checkbox"/>		02/06/2023	12:52:10	13:05:00	0:12:50	770	5	4	0	154.00	192.50
126	<input type="checkbox"/>		02/06/2023	13:10:21	13:26:51	0:16:30	990	8	4	0	123.75	247.50
127	<input type="checkbox"/>		02/06/2023	13:14:12	13:29:46	0:15:34	934	5	3	1	186.80	311.33
128	<input type="checkbox"/>	1	02/06/2023	13:21:59	13:42:57	0:20:58	1,288	8	4	2	157.25	314.50
129	<input type="checkbox"/>	15	02/06/2023	13:26:03	13:51:53	0:25:50	1,550	8	3	3	193.75	516.67
130	<input type="checkbox"/>		02/06/2023	13:29:43	13:48:45	0:19:02	1,142	6	4	3	190.33	285.50
131	<input type="checkbox"/>	19	02/06/2023	13:37:54	13:54:41	0:16:47	1,007	5	4	3	201.40	251.75
132	<input type="checkbox"/>		02/06/2023	13:42:15	14:03:45	0:21:30	1,290	5	4	4	258.00	322.50
133	<input type="checkbox"/>		02/06/2023	13:48:51	14:07:38	0:18:47	1,127	5	4	3	225.40	281.75
134	<input type="checkbox"/>	9	02/06/2023	13:54:45	14:15:23	0:20:38	1,238	8	4	2	154.75	309.50
135	<input type="checkbox"/>		02/06/2023	14:05:04	14:23:49	0:18:45	1,125	9	4	2	125.00	281.25
136	<input type="checkbox"/>	13	02/06/2023	14:13:15	14:34:04	0:20:49	1,249	8	4	2	156.13	312.25
137	<input type="checkbox"/>		02/06/2023	14:23:42	14:54:30	0:30:48	1,848	10	7	2	184.80	264.00
138	<input type="checkbox"/>	12	02/06/2023	14:31:51	14:48:20	0:16:29	989	6	4	2	164.83	247.25
139	<input type="checkbox"/>		02/06/2023	14:36:42	14:53:48	0:17:06	1,026	5	4	2	205.20	256.50
140	<input type="checkbox"/>		02/06/2023	14:43:24	14:58:41	0:15:17	917	6	3	3	152.83	305.67
141	<input type="checkbox"/>		02/06/2023	14:57:54	15:15:09	0:17:15	1,035	5	4	1	207.00	258.75
142	<input type="checkbox"/>		02/06/2023	15:32:45	15:48:41	0:15:56	956	8	3	0	119.50	318.67
143	<input type="checkbox"/>		02/06/2023	15:43:10	15:59:16	0:16:06	966	6	4	1	161.00	241.50
144	<input type="checkbox"/>		02/06/2023	15:48:41	16:04:42	0:16:01	961	8	4	1	120.13	240.25
145	<input type="checkbox"/>		02/06/2023	16:31:52	16:53:01	0:21:09	1,269	10	4	0	126.90	317.25
146	<input type="checkbox"/>		02/06/2023	16:43:42	16:59:16	0:15:34	934	5	4	1	186.80	233.50
147	<input type="checkbox"/>		02/06/2023	19:54:41	20:16:07	0:21:26	1,286	8	5	0	160.75	257.20
148	<input type="checkbox"/>		02/06/2023	20:20:41	20:37:56	0:17:15	1,035	5	4	0	207.00	258.75
149	<input type="checkbox"/>		02/06/2023	20:26:45	20:48:13	0:21:28	1,288	8	5	1	161.00	257.60

Figura B.37: Resultados en crudo de los datos recopilados en el restaurante con el sistema del día 02/06/2023 del experimento 3.

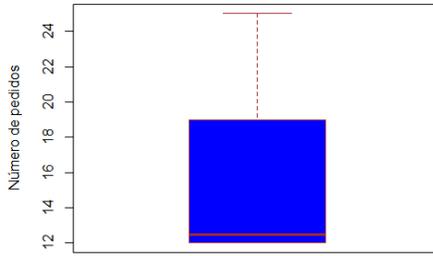


Figura B.39: Diagrama de caja de número de pedidos durante la prueba sin sistema del experimento 3.

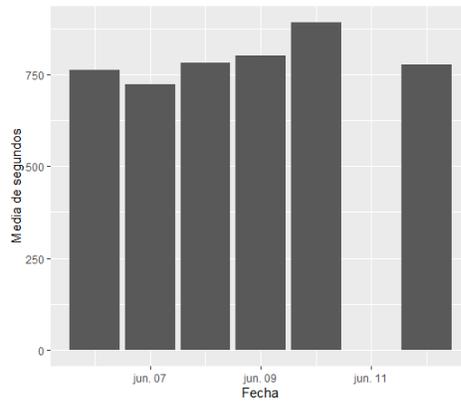


Figura B.40: Gráfico de barras de media de tiempo en segundo que demora un pedido durante la prueba sin sistema del experimento 3.

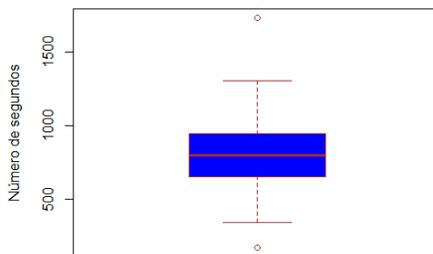


Figura B.41: Diagrama de caja de tiempo en segundo que demora un pedido durante la prueba sin sistema del experimento 3.

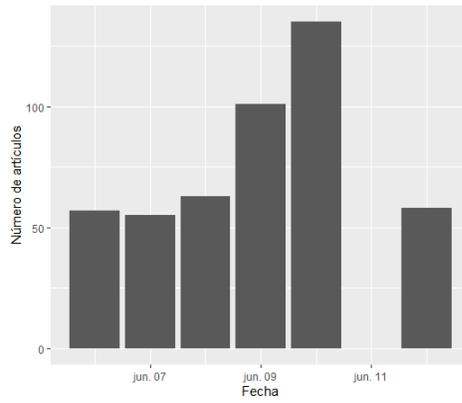


Figura B.42: Gráfico de barras de total de artículos por día durante la prueba sin sistema del experimento 3.

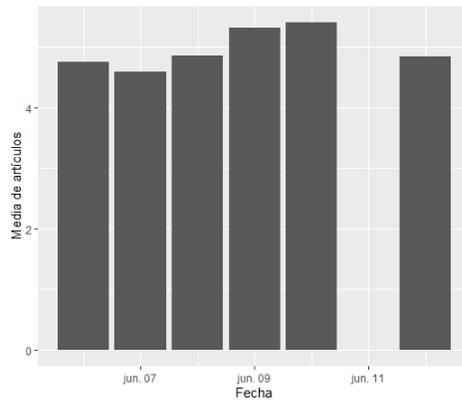


Figura B.43: Gráfico de barras de media de artículos de un pedido por día durante la prueba sin sistema del experimento 3.

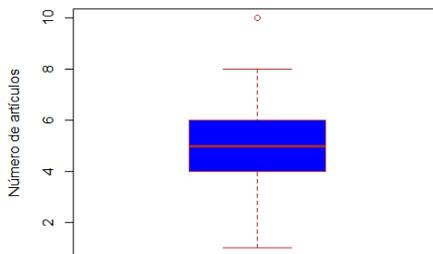


Figura B.44: Diagrama de caja de número de artículos durante la prueba sin sistema del experimento 3.

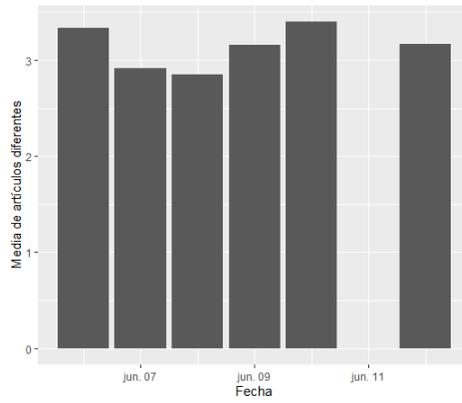


Figura B.45: Gráfico de barras de media de artículos distintos de un pedido durante la prueba sin sistema del experimento 3.

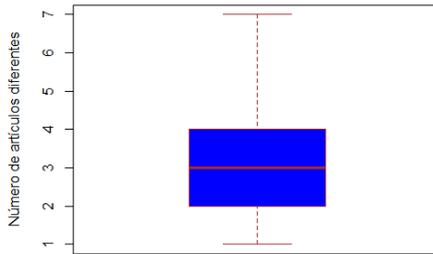


Figura B.46: Diagrama de caja de media de artículos distintos de un pedido durante la prueba sin sistema del experimento 3.

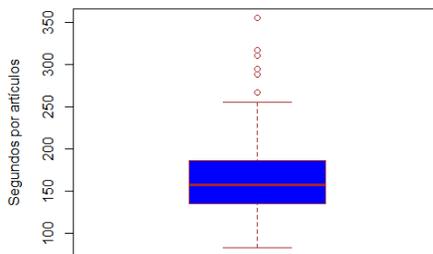


Figura B.47: Diagrama de caja de tiempo en segundos sobre la cantidad de artículos de un pedido durante la prueba sin sistema del experimento 3.

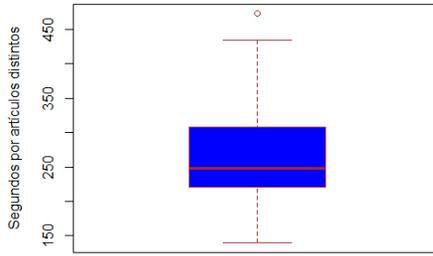


Figura B.48: Diagrama de caja de tiempo en segundos sobre la cantidad de artículos distintos de un pedido durante la prueba sin sistema del experimento 3.

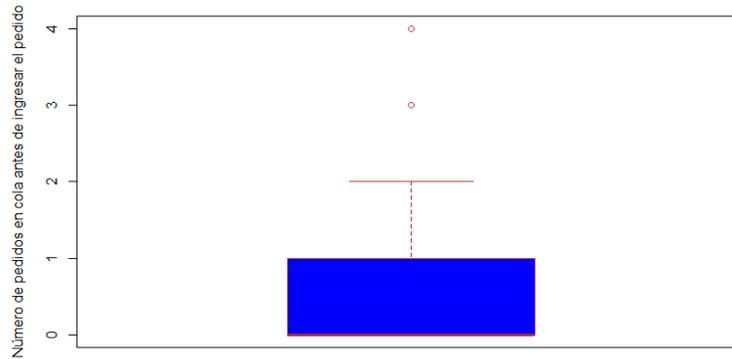


Figura B.49: Diagrama de caja del número de pedidos previamente en cola a la hora del ingreso de un nuevo pedido durante la prueba sin sistema del experimento 3.

Pedido	Programado	Mesa	Dia Ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Diff/Art	Diff/Art dif
1	<input type="checkbox"/>		06/06/2023	12:50:10	13:05:05	0:14:55	895	5	4	0	179.00	223.75
2	<input type="checkbox"/>		06/06/2023	13:15:06	13:25:41	0:10:35	635	2	2	0	317.50	317.50
3	<input type="checkbox"/>		06/06/2023	13:23:16	13:36:13	0:12:57	777	5	4	1	155.40	194.25
4	<input type="checkbox"/>	15	06/06/2023	13:29:41	13:45:53	0:16:12	972	5	4	1	194.40	243.00
5	<input type="checkbox"/>		06/06/2023	14:09:13	14:21:42	0:12:29	749	6	4	0	124.83	187.25
6	<input type="checkbox"/>		06/06/2023	14:23:13	14:29:42	0:06:29	389	3	1	0	129.67	389.00
7	<input type="checkbox"/>		06/06/2023	14:45:56	14:54:51	0:08:55	535	2	2	0	267.50	267.50
8	<input type="checkbox"/>		06/06/2023	14:57:41	15:13:04	0:15:23	923	5	4	0	184.60	230.75
9	<input type="checkbox"/>		06/06/2023	15:40:07	15:50:48	0:10:41	641	4	2	0	160.25	320.50
10	<input type="checkbox"/>		06/06/2023	15:43:01	15:52:51	0:09:50	590	2	2	1	295.00	295.00
11	<input type="checkbox"/>	13	06/06/2023	19:52:17	20:09:45	0:17:28	1,048	10	7	0	104.80	149.71
12	<input type="checkbox"/>		06/06/2023	20:45:51	21:02:24	0:16:33	993	8	4	0	124.13	248.25

Figura B.51: Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 06/06/2023 del experimento 3.

Pedido	Programado	Mesa	Dia Ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Diff/Art	Diff/Art dif
13	<input type="checkbox"/>		07/06/2023	12:31:01	12:42:51	0:11:50	710	4	4	3	177.50	236.67
14	<input type="checkbox"/>		07/06/2023	13:15:18	13:26:48	0:11:30	690	3	3	2	230.00	345.00
15	<input type="checkbox"/>		07/06/2023	13:32:48	13:46:25	0:13:37	817	6	6	2	136.17	488.50
16	<input type="checkbox"/>		07/06/2023	13:47:51	13:58:43	0:10:52	652	5	5	0	130.40	217.33
17	<input type="checkbox"/>	1	07/06/2023	13:57:45	14:11:51	0:14:06	846	4	4	1	211.50	211.50
18	<input type="checkbox"/>	15	07/06/2023	14:15:45	14:31:58	0:16:13	973	8	8	0	121.63	243.25
19	<input type="checkbox"/>		07/06/2023	14:26:42	14:40:01	0:13:19	799	4	4	2	199.75	399.50
20	<input type="checkbox"/>		07/06/2023	14:38:41	14:49:45	0:11:04	664	5	5	1	132.80	221.33
21	<input type="checkbox"/>		07/06/2023	14:53:54	15:08:48	0:14:54	894	5	5	0	178.80	223.50
22	<input type="checkbox"/>		07/06/2023	15:12:41	15:21:05	0:08:24	504	4	4	3	126.00	168.00
23	<input type="checkbox"/>		07/06/2023	18:41:02	18:43:48	0:02:46	166	2	2	1	83.00	166.00
24	<input type="checkbox"/>		07/06/2023	20:14:51	20:31:04	0:16:13	973	5	5	4	194.60	243.25

Figura B.52: Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 07/06/2023 del experimento 3.

Pedido	Programado	Mesa	Dia ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
25	<input type="checkbox"/>		08/06/2023	13:01:51	13:11:08	0:09:17	557	4	4	2	139.25	278.50
26	<input type="checkbox"/>		08/06/2023	13:13:42	13:26:09	0:12:27	747	5	5	3	149.40	249.00
27	<input type="checkbox"/>		08/06/2023	13:26:25	13:38:41	0:12:16	736	5	4	0	147.20	184.00
28	<input type="checkbox"/>	15	08/06/2023	13:34:45	13:50:08	0:15:23	923	5	5	1	184.60	307.67
29	<input type="checkbox"/>	1	08/06/2023	13:55:04	14:10:21	0:15:17	917	6	3	0	152.83	305.67
30	<input type="checkbox"/>		08/06/2023	14:06:08	14:27:12	0:21:04	1264	8	8	5	158.00	252.80
31	<input type="checkbox"/>		08/06/2023	14:26:13	14:38:10	0:11:57	717	4	4	2	179.25	358.50
32	<input type="checkbox"/>		08/06/2023	14:39:53	14:52:41	0:12:48	768	3	3	0	256.00	384.00
33	<input type="checkbox"/>		08/06/2023	14:58:42	15:08:18	0:09:36	576	4	4	2	144.00	288.00
34	<input type="checkbox"/>	9	08/06/2023	15:25:04	15:42:14	0:17:10	1030	6	6	4	171.67	257.50
35	<input type="checkbox"/>		08/06/2023	15:49:30	16:03:01	0:13:31	811	4	4	2	202.75	405.50
36	<input type="checkbox"/>		08/06/2023	18:08:42	18:14:24	0:05:42	342	4	4	2	85.50	171.00
37	<input type="checkbox"/>		08/06/2023	19:44:05	19:56:48	0:12:43	763	5	5	3	152.60	254.33

Figura B.53: Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 08/06/2023 del experimento 3.

Pedido	Programado	Mesa	Día ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Diff/Art	Diff/Art dif
38	<input type="checkbox"/>		09/06/2023	12:25:08	12:40:54	0:15:46	946	10	2	0	94.60	473.00
39	<input type="checkbox"/>		09/06/2023	12:57:16	13:10:51	0:13:35	815	6	2	0	135.83	407.50
40	<input type="checkbox"/>		09/06/2023	13:13:25	13:24:41	0:11:16	676	5	3	0	135.20	225.33
41	<input type="checkbox"/>		09/06/2023	13:21:41	13:32:48	0:11:07	667	5	3	1	133.40	222.33
42	<input type="checkbox"/>		09/06/2023	13:38:04	13:49:56	0:11:52	712	3	3	0	237.33	237.33
43	<input type="checkbox"/>		09/06/2023	13:47:45	13:57:14	0:09:29	569	5	3	1	113.80	189.67
44	<input type="checkbox"/>		09/06/2023	13:59:46	14:12:54	0:13:08	788	5	4	0	157.60	197.00
45	<input type="checkbox"/>		09/06/2023	14:08:35	14:21:05	0:12:30	750	5	3	1	150.00	250.00
46	<input type="checkbox"/>		09/06/2023	14:23:15	14:38:01	0:14:46	886	5	4	0	177.20	221.50
47	<input type="checkbox"/>		09/06/2023	14:34:08	14:51:38	0:17:30	1050	5	4	1	210.00	262.50
48	<input type="checkbox"/>		09/06/2023	14:45:51	15:05:01	0:19:10	1150	10	5	1	115.00	230.00
49	<input type="checkbox"/>		09/06/2023	14:57:35	15:06:08	0:08:33	513	4	2	1	128.25	256.50
50	<input type="checkbox"/>		09/06/2023	15:09:51	15:24:05	0:14:14	854	6	3	0	142.33	284.67
51	<input type="checkbox"/>	15	09/06/2023	15:15:43	15:31:02	0:15:19	919	5	4	1	183.80	229.75
52	<input type="checkbox"/>		09/06/2023	15:23:17	15:34:51	0:11:34	694	4	2	2	173.50	347.00
53	<input type="checkbox"/>		09/06/2023	15:46:08	15:58:41	0:12:33	753	3	2	0	251.00	376.50
54	<input type="checkbox"/>		09/06/2023	20:08:43	20:19:05	0:10:22	622	2	2	0	311.00	311.00
55	<input type="checkbox"/>		09/06/2023	20:26:41	20:40:56	0:14:15	855	5	4	0	171.00	213.75
56	<input type="checkbox"/>		09/06/2023	20:49:05	21:06:04	0:16:59	1019	8	5	0	127.38	203.80

Figura B.54: Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 09/06/2023 del experimento 3.

Pedido	Programado	Mesa	Día Ingreso Y	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
57	<input type="checkbox"/>		10/06/2023	12:31:10	12:48:26	0:17:16	1,036	5	4	0	207.20	259.00
58	<input type="checkbox"/>		10/06/2023	13:03:54	13:12:58	0:09:04	544	4	2	0	136.00	272.00
59	<input type="checkbox"/>	1	10/06/2023	13:06:48	13:19:51	0:13:03	783	5	3	1	156.60	261.00
60	<input type="checkbox"/>		10/06/2023	13:16:46	13:26:04	0:09:18	558	6	4	1	93.00	139.50
61	<input type="checkbox"/>	15	10/06/2023	13:21:08	13:35:28	0:14:20	860	6	2	1	143.33	430.00
62	<input type="checkbox"/>		10/06/2023	13:29:43	13:48:45	0:19:02	1,142	6	4	1	190.33	285.50
63	<input type="checkbox"/>		10/06/2023	13:34:45	13:51:58	0:17:13	1,033	8	5	2	129.13	206.60
64	<input type="checkbox"/>		10/06/2023	13:42:02	13:49:38	0:07:36	456	3	2	2	152.00	228.00
65	<input type="checkbox"/>	9	10/06/2023	13:45:35	13:57:16	0:11:41	701	4	3	3	175.25	233.67
66	<input type="checkbox"/>		10/06/2023	13:48:31	13:59:05	0:10:34	634	2	2	4	317.00	317.00
67	<input type="checkbox"/>		10/06/2023	13:53:15	14:05:27	0:12:12	732	3	3	2	244.00	244.00
68	<input type="checkbox"/>		10/06/2023	14:04:46	14:13:51	0:09:05	545	5	3	1	109.00	181.67
69	<input type="checkbox"/>		10/06/2023	14:15:32	14:37:15	0:21:43	1,303	7	3	0	186.14	434.33
70	<input type="checkbox"/>		10/06/2023	14:23:42	14:54:30	0:30:48	1,848	6	4	1	308.00	462.00
71	<input type="checkbox"/>		10/06/2023	14:27:14	14:48:12	0:20:58	1,258	10	4	2	125.80	314.50
72	<input type="checkbox"/>		10/06/2023	14:36:42	14:53:48	0:17:06	1,026	5	4	3	205.20	256.50
73	<input type="checkbox"/>		10/06/2023	14:39:47	14:57:08	0:17:21	1,041	6	4	3	173.50	260.25
74	<input type="checkbox"/>		10/06/2023	14:50:03	15:01:30	0:11:27	687	5	3	3	137.40	229.00
75	<input type="checkbox"/>		10/06/2023	15:13:20	15:19:15	0:05:55	355	1	1	0	355.00	355.00
76	<input type="checkbox"/>		10/06/2023	15:20:10	15:34:51	0:14:41	881	5	4	0	176.20	220.25
77	<input type="checkbox"/>	9	10/06/2023	15:23:46	15:41:42	0:17:56	1,076	8	8	1	134.50	269.00
78	<input type="checkbox"/>		10/06/2023	16:43:42	16:59:16	0:15:34	934	5	4	0	186.80	233.50
79	<input type="checkbox"/>		10/06/2023	20:20:41	20:33:56	0:13:15	795	5	4	0	159.00	198.75
80	<input type="checkbox"/>	13	10/06/2023	20:38:56	20:55:15	0:16:19	979	7	4	0	139.86	244.75
81	<input type="checkbox"/>		10/06/2023	20:43:43	21:01:53	0:18:10	1,090	8	5	1	136.25	218.00

Figura B.55: Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 10/06/2023 del experimento 3.

Pedido	Programado	Mesa	Dia ingreso	Fecha Ingresado	Fecha termino	Diferencia	segundos	Articulos	Articulos dif	Pedidos antes	Dif/Art	Dif/Art dif
82	<input type="checkbox"/>	1	12/06/2023	12.45.34	12.56.01	0.10.27	627	4	4	2	156.75	313.50
83	<input type="checkbox"/>		12/06/2023	13.03.54	13.12.58	0.09.04	544	4	4	2	136.00	272.00
84	<input type="checkbox"/>	9	12/06/2023	13.14.12	13.29.46	0.15.34	934	5	5	3	186.80	311.33
85	<input type="checkbox"/>		12/06/2023	13.52.07	14.09.36	0.17.29	1.049	6	6	3	174.83	349.67
86	<input type="checkbox"/>		12/06/2023	14.04.51	14.13.51	0.09.00	540	5	5	3	108.00	180.00
87	<input type="checkbox"/>		12/06/2023	14.20.38	14.34.53	0.14.15	855	5	5	4	171.00	213.75
88	<input type="checkbox"/>		12/06/2023	14.53.00	15.06.54	0.13.54	834	4	4	4	208.50	208.50
89	<input type="checkbox"/>		12/06/2023	15.13.05	15.27.24	0.14.19	859	6	6	3	143.17	286.33
90	<input type="checkbox"/>	15	12/06/2023	15.35.01	15.51.35	0.16.34	994	5	5	4	198.80	248.50
91	<input type="checkbox"/>		12/06/2023	17.35.24	17.43.15	0.07.51	471	4	4	2	117.75	235.50
92	<input type="checkbox"/>		12/06/2023	19.31.35	19.45.24	0.13.49	829	5	5	4	165.80	207.25
93	<input type="checkbox"/>		12/06/2023	20.35.47	20.49.05	0.13.18	798	5	5	4	159.60	199.50

Figura B.56: Resultados en crudo de los datos recopilados en el restaurante sin el sistema del día 12/06/2023 del experimento 3.