

UNIVERSIDAD POLITÉCNICA DE PUEBLA
Ingeniería en Informática



**Proyecto de Estancia Práctica en
Desarrollador en Sistemas de Software y
Administrador de Redes**

“Desarrollo de un software que ayude a la administración de materiales y pedidos del restaurante “La Ventanita””

Área temática del CONACYT: VII
Ingenierías y tecnologías

Presenta:

Miguel Angel Cuaya Tepeyahuitl

Asesor técnico

Lic. Karen Mónica Simbro García

Asesor académico

MC. Rebeca Rodríguez Huesca

Juan C. Bonilla, Puebla, México.

18 de Diciembre de 2020

Resumen

En el presente documento se describirá el proceso que se llevó a cabo a lo largo del periodo septiembre-diciembre, para el desarrollo de un software de escritorio como producto del periodo de prácticas llamado Estancia II.

A lo largo de mi estancia II se realizó el proyecto de Desarrollo de un software que ayude a la administración de materiales y pedidos del restaurante “La Ventanita”. Las actividades desarrolladas se resumen en análisis de la situación que se vivía en el restaurante “La Ventanita” referente a la administración, diseño y realización de una base de datos, desarrollo de un software de escritorio y elaboración de un plan de pruebas.

Se describirá cada una de las herramientas utilizadas y se justificará el motivo por el cual se decidió que el desarrollo del software fuera en el lenguaje de programación JAVA en la IDE (entorno de desarrollo integrado) NetBeans, junto con el gestor de base de datos de MySQL. En adición a lo anterior se describirán los desafíos encontrados durante las 240 horas y como se le dio solución a los mismos, siguiendo los pasos de la metodología de modelo de prototipos.

El documento se compone de 5 capítulos: en el capítulo 1 (introducción), se expondrá la problemática, así como el objetivo general y específicos, en el capítulo 2 (metodología), se describirá cada una de las fases de la metodología seleccionada para el desarrollo del proyecto y las herramientas utilizadas, en el capítulo 3 (resultados) se mostrarán los resultados obtenidos de cada una de las etapas de desarrollo, en el capítulo 4 (conclusiones y recomendaciones) se contarán las conclusiones, experiencias y enseñanzas adquiridas. Para finalizar el capítulo 5 (Referencias bibliográficas), se indicarán las fuentes de información consultadas. Con el sistema desarrollado se pretende ayudar a los administrativos del restaurante a tener un mejor manejo de información.

Índice

| | |
|--|-----|
| 1. Introducción | 4 |
| 1.1 Descripción del problema o necesidad | 4 |
| 1.2 Justificación | 5 |
| 1.3 Objetivo General y Específicos | 6 |
| 2. Metodología y herramientas | 8 |
| 2.1 Descripción de la metodología | 8 |
| 2.2 Herramientas tecnológicas | 12 |
| 3. Resultados | 19 |
| 3.1 Recolección y refinamiento de requisitos | 19 |
| 3.2 Modelado o diseño rápido | 26 |
| 3.3 Construcción del prototipo | 33 |
| 3.4 Desarrollo y evaluación del prototipo por el cliente | 50 |
| 3.5 Refinamiento del prototipo | 52 |
| 3.6 Producto Final | 64 |
| 3.7 Plan de pruebas | 65 |
| 4. Conclusiones y recomendaciones | 118 |
| 5. Referencias bibliográficas | 119 |

1. Introducción

Día con día los dispositivos electrónicos van tomando mayor importancia en la vida de los seres humanos haciendo que una tarea fundamental de la informática sea la automatización de procesos, en otras palabras, hacer más con menos acciones.

A continuación, se describirá la problemática que sitúa el restaurante “La Ventanita”. Así mismo, se expondrá la justificación, objetivo general y específicos con el fin de dar a conocer el motivo de la realización del sistema.

1.1 Descripción del problema o necesidad

Desde su fundación en el año 2009, instalaciones de malla ciclónica MASSCO, ha mantenido como objetivo fundamental la seguridad y calidad de los servicios que ofrece, los cuales son: instalación y elaboración de malla ciclónica, cancelas, ventanales, puertas, ventanas, entre otros muchos productos relacionados con cristalería, herrería y aluminio.

Recientemente debido a un deseo de expansión y de llegar a un público más amplio, realizaron la apertura de un restaurante llamado “LA VENTANITA” (Figura 1)



Figura 1. Logotipo de restaurante – bar “La Ventanita”

Debido a la reciente apertura que tiene el restaurante “La Ventanita”, éste cuenta con dificultades para realizar una buena gestión, ya que ésta es muy arcaica; dicho procedimiento se realiza haciendo el uso de una libreta, es decir, los pedidos, materiales de inventario, notas, entre otros datos se registran en ella.

Lo anterior genera 2 problemas principales: seguridad y fiabilidad de información.

- **Seguridad:** Al contar con un sistema que no posee un login, contraseña o algún método de seguridad, existe la posibilidad de que llegue un tercero a alterar datos de cuentas, notas, y/o cantidad de productos en inventario con relativa facilidad, lo que genera el segundo problema.
- **Falta de fiabilidad de la información:** La gerente del restaurante al no ser la única persona que tiene alcance a la libreta en donde se realizan registros del restaurante le resulta difícil tener un control exacto de quien agrega o modifica información, lo que al final del día genera dudas al desconocer si la información de pedidos, materiales de inventario, notas a meseros, entre otros datos, son datos confiables o fueron alterados de alguna forma. Su sistema de gestión se presta a pérdida, duplicidad y falta de control de su información.

1.2 Justificación

Con el continuo crecimiento que va teniendo la informática ahora es posible hacer múltiples tareas con una sola herramienta.

Con base en lo anterior, se propone el desarrollo de un software de escritorio que ayude a administrar de manera eficiente los procesos de gestión de inventario, almacenamiento, modificación de pedidos, notas para meseros y materiales en una base de datos.

El desarrollo del software de escritorio acortará tiempos de consulta, además de permitir a la gerente poder visualizar la información del inventario existente, costo de platillos y pedidos, almacenados en la base de datos. Mediante diferentes interfaces dedicadas a cada uno de los apartados (pedidos, costo de platillos y materias primas) se permitirá que las consultas y modificación de información sea fácil de utilizar.

En el sistema existirán dos tipos de usuarios, el “usuario normal” y el “usuario administrador”. El “usuario normal” sólo podrá consultar el inventario, crear y modificar pedidos, mientras que el “usuario administrador” tendrá acceso a cada uno de los apartados del sistema. Siendo el gerente el “usuario administrador”, podrá modificar absolutamente todo referente a datos almacenados, además de la posibilidad de agregar más “usuarios normales”. Para distinguir a los dos tipos de usuarios se agregará una interfaz que sirva como login en donde dependiendo del usuario que se loguee se le mostraran las interfaces correspondientes.

La finalidad del login es reforzar la seguridad del sistema y que no cualquier usuario tenga acceso a todos los apartados del sistema.

A grandes rasgos el sistema brindará una herramienta que ayude a la administración y seguridad de datos del restaurante “La Ventanita”, permitiendo una mejor administración y control de información haciendo uso de herramientas de nueva generación.

El impacto que se busca es acercar cada vez más a las personas a esta nueva era informática, de forma que conozcan y se familiaricen con las nuevas tecnologías.

Este proyecto se desarrollará en el lenguaje Java, teniendo en cuenta su fácil implementación y manejo.

Se hará uso del lenguaje de programación JAVA, el gestor de base de datos de MySQL, así como los programas Netbeans (12.0), phpMyAdmin y el sitio web lucidchart.com.

1.3 Objetivo General y Específicos

Objetivo general:

- Desarrollar un software que permita al usuario administrar su inventario y gestionar de una mejor forma los pedidos del restaurante “La Ventanita”

Objetivos específicos:

- Analizar los requerimientos que tiene el restaurante “La Ventanita” respecto a la administración de pedidos, seguridad y administración.

- Diseñar una base de datos relacional e interfaces a implementar con base al análisis de los requerimientos del restaurante “La Ventanita”.
- Programar un software de escritorio que permita las acciones de creación, consulta, actualización y eliminación de pedidos y materiales para el restaurante “La Ventanita”.
- Elaborar un plan de pruebas de caja negra alusivo a casos de uso sobre las funciones principales del proyecto que permita conocer si el software cumple con los requerimientos del restaurante “La Ventanita”

2. Metodología y herramientas

Existen varios modelos, paradigmas y filosofías de desarrollo, en los cuales se apoya la ingeniería de software para la construcción del software.

En este capítulo se definirá: metodología, herramientas tecnológicas y lenguajes utilizados a lo largo del desarrollo del proyecto para el restaurante “La Ventanita”, con el fin de ampliar el contexto del mismo además de demostrar porqué es la solución óptima a la problemática planteada

2.1 Descripción de la metodología

La metodología de modelo de prototipos pertenece a los modelos de desarrollo evolutivo, permite que un sistema se construya rápidamente para comprender con facilidad aspectos en los que se asegure que el desarrollador, cliente o usuario final estén de acuerdo en lo que se necesita, así como la solución que se propone para dicha necesidad; de esta forma minimizar riesgos e incertidumbre en el desarrollo. Este modelo se aplica cuando el cliente define un conjunto de objetivos generales para el software a desarrollar sin determinar detalladamente los requisitos de entrada y salida, dicho de otra forma, cuando no se está seguro de la adaptabilidad del sistema o de la forma en que interactúa el hombre y la máquina. [9]

2.1.1 Etapas del modelo de prototipos: (Figura 2)

2.1.1.1 Recolección y refinamiento de requisitos

Se analizan las necesidades del usuario final del software, se hace la recolección de requisitos, una planificación de las actividades a realizar y el cliente define los objetivos globales a cubrir.

2.1.1.2 Modelado o diseño rápido

Se realiza un diseño rápido de una representación de aquellos aspectos del software que serán visibles para el cliente o usuario final.

El diseño conduce a la construcción de un prototipo

2.1.1.3 Construcción del prototipo

Para muchas interfaces de usuario, este modelo puede resultar tan simple como unos dibujos con lápiz y papel o tan complejo como el propio código operativo final. Para interfaces de hardware o estaciones de trabajo, el modelo puede consistir en maquetas de espuma, caucho, cartón o cartulina. Cuanto más próximo se encuentre el prototipo al producto real, mejor será la evaluación; se pueden obtener magníficos resultados con prototipos de baja fidelidad. [1]

Se comienza elaborando un prototipo del producto final: qué aspecto tendrá y cómo funcionará. Los prototipos deben ser construidos en poco tiempo, usando los programas adecuados teniendo en cuenta no utilizar muchos recursos.

2.1.1.4 Desarrollo y evaluación del prototipo por el cliente

Los diseños son analizados y evaluados por el cliente o usuarios, lo que permite concretar y refinar los requisitos del software a desarrollar.

2.1.1.5 Refinamiento del prototipo

Se modifica el prototipo de acuerdo a la información obtenida en el paso anterior para que satisfaga las necesidades del cliente, al tiempo que facilita al ingeniero de software un mejor conocimiento del sistema. El prototipo ha de abarcar, cada vez más las características del producto final. Llegados a un punto, la construcción de prototipos adicionales resultará menos eficiente que usar las construcciones iniciales para el producto. [1]

2.1.1.6 Producto Final

Se llega a este paso cuando todas las etapas anteriores fueron cumplidas satisfactoriamente. Haciendo que cliente este satisfecho con el prototipo final.



Figura 2. Etapas del modelo de prototipos

2.1.2 Tipos de prototipo [2]:

2.1.2.1 Rápido (Rapid Prototyping)

Metodología de diseño que desarrolla rápidamente nuevos diseños, los evalúa y prescinde del prototipo cuando el próximo diseño es desarrollado mediante un nuevo prototipo.

2.1.2.2 Reutilizable (Reusable Prototyping)

"Evolutionary Prototyping"; no se pierde el esfuerzo efectuado en la construcción del prototipo pues sus partes o el conjunto pueden ser utilizados para construir el producto real.

2.1.2.3 Modular (Modular Prototyping)

También conocido como Prototipado Incremental (Incremental prototyping); se añaden nuevos elementos sobre el prototipo a medida que el ciclo de diseño progresa.

2.1.2.4 Horizontal (Horizontal Prototyping)

El prototipo cubre un amplio número de aspectos y funciones, pero la mayoría no son operativas. Resulta muy útil para evaluar el alcance del producto, pero no su uso real.

2.1.2.5 Vertical (Vertical Prototyping)

El prototipo cubre sólo un pequeño número de funciones operativas. Resulta muy útil para evaluar el uso real sobre una pequeña parte del producto.

2.1.2.6 Alta-fidelidad (High-fidelity Prototyping)

El prototipo se implementa de la forma más cercana posible al diseño real en términos de aspecto, impresiones, interacción y tiempo.

2.1.2.7 Baja-fidelidad (Low-fidelity Prototyping)

Los prototipos de baja fidelidad a menudo se basan en papel y no permiten interacciones del usuario. Van desde una serie de maquetas dibujadas a mano hasta impresiones. En teoría, los bocetos de baja fidelidad son más rápidos de crear. Los prototipos de baja fidelidad son útiles para permitir la visualización temprana de soluciones de diseño alternativas, lo que ayuda a provocar innovación y mejora. Una ventaja adicional de este enfoque es que al utilizar bocetos preliminares, los usuarios pueden sentirse más cómodos sugiriendo cambios.

2.1.3 Ventajas y desventajas

2.1.3.1 Ventajas

- No modifica el flujo del ciclo de vida

- Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios
- Reduce costo y aumenta la probabilidad de éxito
- Exige disponer de las herramientas adecuadas
- Ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo.

2.1.3.2 Desventajas

- Debido a que el usuario ve que el prototipo funciona piensa que este es el producto terminado
- El desarrollador puede caer en la tentación de ampliar el prototipo para construir el sistema final sin tener en cuenta los compromisos de calidad y mantenimiento que tiene con el cliente.

2.2 Herramientas tecnológicas

A continuación, se hace mención de las herramientas utilizadas.

2.2.1 Lenguaje de programación Java

Java es un lenguaje de programación y una plataforma informática. Se caracteriza por ser rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta computadoras avanzadas, desde teléfonos móviles hasta Internet, Java está en todas partes, si es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. [5]

2.2.1.1 Características [4]

- **Simplicidad:** Java ofrece la funcionalidad de un lenguaje potente, derivado de C y C++, pero sin las características menos usadas y más confusas de estos, haciéndolo más sencillo.
- **Orientado a objetos:** Permite diseñar el software de forma que los distintos tipos de datos que se usen estén unidos a sus operaciones.

- **Independiente a la plataforma:** Esto significa que programas escritos en el lenguaje Java pueden ejecutarse en cualquier tipo de hardware, lo que lo hace portable.
- **Recolector de basura:** Cuando no hay referencias localizadas a un objeto, el recolector de basura de Java borra dicho objeto, liberando así la memoria que ocupaba. Esto previene posibles fugas de memoria.
- **Es seguro y sólido:** Proporcionando una plataforma segura para desarrollar y ejecutar aplicaciones que, administra automáticamente la memoria, provee canales de comunicación segura protegiendo la privacidad de los datos y, al tener una sintaxis rigurosa evita que se quiebre el código, es decir, no permite la corrupción del mismo.
- **Es multi-hilo:** Java logra llevar a cabo varias tareas simultáneamente dentro del mismo programa. Esto permite mejorar el rendimiento y la velocidad de ejecución.

2.2.1.2 Ventajas [4]

- **Multiplataforma.** No necesitas conocer a priori el tipo de ordenador o el sistema operativo para el que estás programando. Puedes ejecutar EL MISMO programa en un PC con Windows, otro con Linux, en un Servidor SUN con sistema operativo Solaris, o en un teléfono móvil de última generación.
- **Orientado a objetos.** Capaz de acercar la forma de programar a la forma de pensar del ser humano.
- **Liberación de memoria en el sistema:** Java rompe con el sistema tradicional de liberación de memoria, haciendo que el programador ya no fuese el responsable de esa tarea.
- **Fácil de aprender.**
- **Librerías Estándar:** Java viene acompañado de una serie de librerías estándar para realizar multitud de operaciones.
- **Variedad de editores (IDEs):** que aportan multitud de ayudas a la programación, haciendo que el desarrollo sea más fluido y cómodo.

- **Try y catch o excepciones.** En C o C++ no existe un mecanismo específico para la gestión de los errores que puedan producirse en el código.

2.2.1.3 Desventajas [4]

- **Rendimiento:** En la ejecución de programas suele ser un poco menor
- Sólo podemos ejecutar un programa en Java si disponemos de una máquina virtual (JVM)
- No es recomendado para personas nuevas en informática
- Su sintaxis comparada con C# o Python parece para algunos bastante engorrosa y al contrario que su semejante en .NET, C#, es un lenguaje que evoluciona muy lentamente.

2.2.2 phpMyAdmin

Herramienta de administración de código abierto para MySQL y MariaDB. Como aplicación web portátil escrita principalmente en PHP, se ha convertido en una de las herramientas de administración de MySQL más populares.

2.2.2.1 Características [6]

- Interface Web
- Manejador de base de datos MySQL, MariaDB y Drizzle
- Importación de datos desde CSV y SQL
- Exporta datos a varios formatos: CSV, SQL, XML, PDF (via la biblioteca TCPDF), ISO/IEC 26300 - OpenDocument Text y Spreadsheet, Word, Excel, LaTeX y otros
- Administración de múltiples servidores
- Crea gráficos PDF del diseño de la base de datos
- Crea consultas complejas usando Query-by-Example (QBE)
- Búsqueda global en una base de datos o un subconjunto de esta
- Transforma datos almacenados a cualquier formato usando un conjunto de funciones predefinidas, tal como BLOB

- Live charts para monitoriar las actividades del servidor MySQL tales como conexiones, procesos, uso de CPU/Memoria, etc.

2.2.2.2 Ventajas

- Posee una interfaz web intuitiva
- Sirve de referencia para la creación de phpgadmin
- Se encuentra bajo licencia gnu gpl que nos permite la libre distribución, modificación y uso.
- Se pueden importar datos de archivos cvs y sql.

2.2.2.3 Desventajas

- Limita la cantidad que maneja de bases de datos
- Solo administra bases de datos en MySQL

2.2.3 NeatBeans

Entorno de desarrollo integrado (IDE) para Java, permite que las aplicaciones se desarrollen a partir de un conjunto de componentes de software modulares llamados módulos. Además del desarrollo de Java, tiene extensiones para otros lenguajes como PHP, C, C ++, HTML5, y JavaScript.

2.2.3.1 Características

- Sistema basado en Ant, control de versiones y refactoring.
- Esta plataforma incluye servicios para el control del interfaz de usuario, la configuración, el almacenamiento, las ventanas, etc.
- El IDE está desarrollado con la misma metodología modular, por lo que puede extenderse incluyendo módulos con funcionalidades determinadas. [7]

2.2.3.2 Ventajas

- Lenguaje Multi-plataforma: El código que es escrito en java es leído por un intérprete, por lo que su programa andará en cualquier plataforma.

- Manejo automático de la memoria. (para los que vienen de C/C++). El manejo de la memoria se hace automáticamente y utilizando el garbage collector.
- ¡Es gratis!
- Desarrolla aplicaciones web dinámicas.

2.2.3.3 Desventajas

- Lentitud a la hora de ejecutar las aplicaciones
- Requiere un intérprete.
- Algunas implementaciones y librerías pueden tener código rebuscado.
- Una mala implementación de un programa en java, puede resultar en algo muy lento.
- Algunas herramientas tienen un costo adicional

2.2.4 Microsoft Paint

Sencillo editor de gráficos que se incluye con todas las versiones de Microsoft.

2.2.4.1 Características

Abre y guarda archivos en formato de mapa de bits de Windows (BMP), JPEG, GIF, PNG, etc.

El programa puede estar en modo de color o en blanco y negro. Por su simplicidad, se convirtió en una de las aplicaciones más utilizadas en las primeras versiones de Windows, introduciendo a muchos a pintar en una computadora por primera vez. Todavía se usa ampliamente para tareas simples de manipulación de imágenes [10]

2.2.4.2 Ventajas

- **Programa muy simple:** No requiere de gran destreza para ser utilizado.
- **Ediciones rápidas:** debido a su limitado listado de herramientas.

2.2.4.3 Desventajas

- **No variedad de herramientas:** No cuenta con un amplio catálogo de herramientas, por lo que solo se usa para trabajos de edición simples.

2.2.5 MySQL

Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server [8]

2.2.5.1 Características

El segundo más usado detrás de Oracle según DB-Engines Ranking.

Una base de datos. Fue escrito en C y C++, probado en un amplio número de compiladores, proporciona sistemas de almacenamiento transaccionales y no transaccionales. Su uso, así como su comprensión es relativamente sencillo. [8]

2.2.5.2 Ventajas

- **Sistema de administración relacional:** cuenta con un esquema relacional de bases de datos.
- **Velocidad y flexibilidad:** Dado que es relacional, archiva datos en tablas separadas en lugar de colocar todos los datos en un gran archivo lo que hace más fácil consultas y modificaciones
- **Tablas conectadas:** Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas.
- **Es de fuente abierta:** Significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades.
- **Se mantiene en constante evolución:** pero el núcleo de SQL cambia muy poco: una vez que aprendas su funcionamiento solo tienes que concentrarte en las innovaciones posteriores.

2.2.6 ¿Por qué todas estas herramientas?

Las herramientas elegidas van de la mano con la metodología elegida. Se tuvo que elegir minuciosamente cada uno de los programas a emplear teniendo en cuenta que tenían que ser de código abierto (open source).

Para poder ejecutar el lenguaje de programación Java se necesita de una IDE que nos ayude a una buena codificación; NetBeans es una herramienta que nos brinda ayuda en la codificación.

Además de eso se necesitaba de un gestor de base de datos para MySQL. phpMyAdmin fue el gestor elegido pues al brindarnos herramientas que ayudan a la visualización de la estructura de la base de datos, hace que el diseño e implementación de la misma sea más fácil y ágil.

Paint, el editor de imágenes Microsoft que nos ayuda al diseño y elaboración de los fondos de las interfaces a emplear.

3. Resultados

Se describirán los pasos seguidos desde la especificación hasta la finalización del proyecto. Aquí se plantean cada una de las decisiones tomadas y da mayor fundamentación y justificación a las decisiones tomadas de manera que se pueda comprender el por qué se decidió realizar cada acción. Adentraremos más a las pruebas realizadas y conoceremos a detalle el producto final.

3.1 Recolección y refinamiento de requisitos

Se identificaron los requerimientos que fueron necesarios para el desarrollo del software. Al tener claro los requisitos del restaurante resulta sencillo implementar funcionalidades al sistema que ayuden a mitigar los problemas existentes.

Historias de usuario

Para la recolección de requisitos se usó el método de “historias de usuario”, pues al ser un método que se usa principalmente en metodologías ágiles es ideal para la metodología propuesta. Se eligió este método porque puede determinar requisitos pequeños y específicos, incluida información básica y objetiva detallada por el cliente con un lenguaje coloquial.

Para generar las “historias de usuarios” se utilizó el método de observación en campo, por el cual se obtuvo la siguiente conclusión: “El restaurante “La Ventanita” no cuenta con un sistema de gestión y punto de venta que le ayude a la administración de su negocio y facilite conocer rápidamente información de sus empleados, inventario, órdenes, así como sus ingresos.”.

A continuación, se mostrarán las historias de usuario.

Gerente

1. Login Gerente

Yo como: gerente, **quiero:** poder ingresar al menú de gerentes con mi correo electrónico y con contraseña **para:** poder hacer uso del sistema.

Criterios de aceptación:

- ✓ El gerente ingresa al menú “Gerente” del sistema.
- ✓ El gerente debe estar previamente registrado en el sistema de lo contrario no podrá acceder a el.
- ✓ Solo los usuarios con perfil de gerente podrán acceder al menú para gerentes

2. Agregar nuevos datos de usuario.

Yo como: gerente, **quiero:** poder agregar información nueva sobre usuarios **para:** posteriormente asociarlos a un rol de mesero o gerente.

Criterios de aceptación:

- ✓ Los datos son almacenados en la base de datos.
- ✓ Si un campo del formulario esta sin llenar no se almacena la información.
- ✓ Si se ingresa un correo ya existe en la base de datos, el registro no se realizará.

3. Modificar datos de usuarios.

Yo como: gerente, **quiero:** poder consultar información de datos almacenados **para:** modificar algún apartado si es necesario.

Criterios de aceptación:

- ✓ Se muestra en la interfaz un botón para poder buscar la información de un usuario.
- ✓ Se visualiza en una ventana los apartados de información de un usuario previamente registrado.
- ✓ Se modifican los datos almacenados de un usuario.
- ✓ El sistema muestra un mensaje en pantalla que confirma que los datos fueron modificados correctamente.

4. Eliminar a usuarios sin un rol.

Yo como: gerente, **quiero:** eliminar datos de un usuario sin un rol **para:** no contar con información sin usar.

Criterios de aceptación:

- ✓ Se borra la información de manera exitosa.
- ✓ Solo se puede borrar el registro de información de un usuario sin un rol en el sistema.

5. Asociar datos de un usuario sin un rol a un puesto.

Yo como: gerente, **quiero:** asociar datos de un usuario **para:** darle un rol dentro del sistema.

Criterios de aceptación:

- ✓ La asociación de rol queda almacenada correctamente.
- ✓ No se pueden asociar datos que ya cuentan con un rol.
- ✓ Se muestra un mensaje en pantalla de que la asociación de datos ha sido exitosa

6. Modificar rol de usuario.

Yo como: gerente, **quiero:** modificar el registro del rol de un usuario **para:** quitarle o agregarle permisos en el software.

Criterios de aceptación:

- ✓ El rol es cambiado satisfactoriamente.
- ✓ No se puede cambiar un rol si el usuario ya tiene un registro previo (órdenes, inventario o mensajes) antes del intento de cambio.
- ✓ Se muestra un mensaje en pantalla de que el registro ha sido cambiado.

7. Eliminar registro de asociación de información a un rol.

Yo como: gerente, **quiero:** borrar el registro de asociación de datos **para:** poder borrar la información de ese usuario.

Criterios de aceptación:

- ✓ El registro es eliminado correctamente si no se cuenta con una acción (en órdenes, inventario o mensajes) dentro del sistema.

- ✓ Se muestra mensaje en pantalla de que se ha borrado el registro de rol.

8. Contar con botón de búsqueda.

Yo como: gerente **quiero:** ver los identificadores en un botón de búsqueda

para: acortar tiempos de búsqueda.

Criterios de aceptación:

- ✓ En cada interfaz de consulta, modificación o asociación se contará con un botón que facilite la búsqueda de registros al usuario.
- ✓ En el botón aparecen los identificadores de registros existentes.
- ✓ Al seleccionar el identificador deseado se llenarán los formularios manipulables.

9. Agregar productos al inventario.

Yo como: gerente **quiero:** agregar un nuevo producto al inventario **para:** tener un registro de lo que tengo en mi almacén de materias primas.

Criterios de aceptación:

- ✓ Se agrega el producto a la base de datos siempre y cuando no exista un registro con el mismo nombre.
- ✓ Para poder agregar un nuevo producto se tiene que especificar la cantidad, a que categoría pertenece y el tipo de unidad de medición del producto.
- ✓ Se muestra un mensaje en pantalla de registro exitoso.

10. Modificar cantidad de producto de inventario.

Yo como: gerente, **quiero:** modificar la cantidad de un producto existente en mi inventario **para:** tener un control más exacto.

Criterios de aceptación.

- ✓ La modificación se realiza satisfactoriamente mostrando un mensaje en pantalla que lo especifique.
- ✓ Para hacer la modificación primero se debe conocer el identificador del producto a modificar.

11. Ver tabla con todos los productos de inventario.

Yo como gerente, **quiero:** ver una tabla que me muestre todos los productos de inventario **para:** poder conocer información acerca del inventario

Criterios de aceptación:

- ✓ Se visualiza una tabla con el nombre, cantidad, fecha de modificación y quien realizo el registro de un producto del inventario.

12. Mensajes

Yo como: gerente, **quiero:** guardar un mensaje que posteriormente verán los usuarios de tipo mesero **para:** tener un medio de comunicación en donde no tenga que interactuar directamente con el remitente.

Criterios de aceptación:

- ✓ El mensaje es almacenado correctamente mostrando un mensaje en pantalla.

13. Ver una tabla en donde se muestren todos los mensajes almacenados de una fecha determinada.

Yo como: gerente, **quiero:** visualizar una tabla en donde seleccione una fecha determinada **para:** poder conocer que mensaje se guardó en la fecha seleccionada.

Criterios de aceptación:

- ✓ La interfaz de consulta se visualiza un botón que muestra las fechas en donde se almacenaron mensajes previamente.
- ✓ Al seleccionar la fecha de consulta deseada se mostrarán todos los mensajes realizados en ese día.

14. Agregar productos a mi registro de menú.

Yo como: gerente, **quiero:** agregar platillos a un registro de orden **para:** poder utilizar la información en transacciones futuras.

Criterios de aceptación:

- ✓ El platillo nuevo es almacenado correctamente siempre y cuando todos los apartados del formulario sean llenados satisfactoriamente.

- ✓ Se muestra un mensaje que confirma que el registro se hizo correctamente.

15. Modificar nombre y precio de un producto de mi registro menú.

Yo como: gerente, **quiero:** modificar el nombre grupo y unidad de medida de mi registro de platillos **para:** tener mi registro actualizado.

Criterios de aceptación:

- ✓ Los datos son modificados correctamente mostrando un mensaje en pantalla que lo confirme.
- ✓ Mi modificación se ve reflejada en mi tabla donde visualizo todos mis platillos.

16. Eliminar registro de un platillo

Yo como: gerente, **quiero:** eliminar el registro de un platillo **para:** ya no mostrarlo en mi registro de platillos.

Criterios de aceptación:

- ✓ El producto se puede borrar siempre y cuando no exista ya una orden con el mismo.
- ✓ El platillo ya no se muestra en el registro de platillos

17. Cobrar orden

Yo como: gerente, **quiero:** cobrar una orden que ha hecho mi mesero **para:** conocer el total de ganancias de la orden en cuestión.

Criterios de aceptación:

- ✓ La interfaz de cobro muestra el total de la orden y modifica el total del registro.
- ✓ Se muestra un mensaje que confirma el cobro.

18. Consultar ganancias de un día determinado

Yo como: gerente, **quiero:** conocer las ganancias registradas de un día en específico **para:** poder como está la situación del restaurante respecto a ingresos.

Criterios de aceptación:

- ✓ El sistema muestra la suma de todas las cuentas cobradas del día seleccionado.
- ✓ Para que una orden sea contemplada en este conteo se tiene que cobrar primero.

Meseros:

1. Consultar inventario.

Yo como mesero, **quiero:** ver una tabla que me muestre todos los productos de inventario **para:** conocer información de existencia de materias primas del restaurante.

Criterios de aceptación:

- ✓ Se muestra una tabla con el nombre, cantidad, fecha de modificación y quien realizo el registro de un producto del inventario.

2. Generar registro de orden.

Yo como: mesero, **quiero:** generar un nuevo registro de orden **para:** poder tomar el pedido de un comensal.

Criterios de aceptación:

- ✓ El registro se lleva acabo satisfactoriamente mostrando un mensaje en pantalla que lo confirme.
- ✓ El número identificador de la orden aparece en el apartado donde se agregan platillos a las órdenes.

3. Agregar platillos a una orden

Yo como: mesero, **quiero:** agregar platillos a una orden **para:** almacenar información de lo que se pidió.

Criterios de aceptación:

- ✓ Se muestra un mensaje de confirmación de platillo agregado.
- ✓ En el registro de la orden se visualizan todos los platillos asociados a ella.

4. Ver mensajes de gerentes

Yo como: mesero, **quiero:** consultar los mensajes que un gerente previamente ha almacenado **para:** conocer el contenido de los mismos.

- ✓ Se visualiza una interfaz con un botón en donde se mostrarán las fechas en donde se realizaron mensajes.
- ✓ Se muestran los mensajes de la fecha elegida.

Los casos de usuario me sirvieron para identificar concretamente lo que él usuario busca con el software.

3.2 Modelado o diseño rápido

Se diseñó la base de datos e interfaces que posteriormente se implementaron.

Base de datos

La base de datos deberá almacenar los siguientes datos:

- Datos: Nombre completo, teléfono, identificador, email y contraseña.
- User: Identificador de usuario, identificador de datos, Nivel (puesto).
- Mensaje: Identificador, identificador del usuario que realizo, fecha.
- M_primas: Identificador, fecha, nombre del producto, cantidad, Identificador de unidad de medición, identificador de categoría, identificador de usuario que realizo la modificación.
- Unit: Identificador, descripción de unidad.
- Category: Identificador, descripción de la unidad.
- Ordenes: Identificador, identificador de usuario, fecha, nota para cocina y total de cuenta.
- Ordenes_Platillos: Identificador de orden e identificador de platillo, esta tabla es meramente para hacer relaciones de muchos a muchos, pues para un pedido es difícil conocer exactamente cuántos platillos pedirá.
- Platillos: Identificador, descripción, precio e identificador de grupo.
- Grupo: Identificador y descripción. (tabla que me ayuda a almacenar las diferentes categorías de productos que existen en la carta tipo menú).

Diagrama de la base de datos (Figura 3)

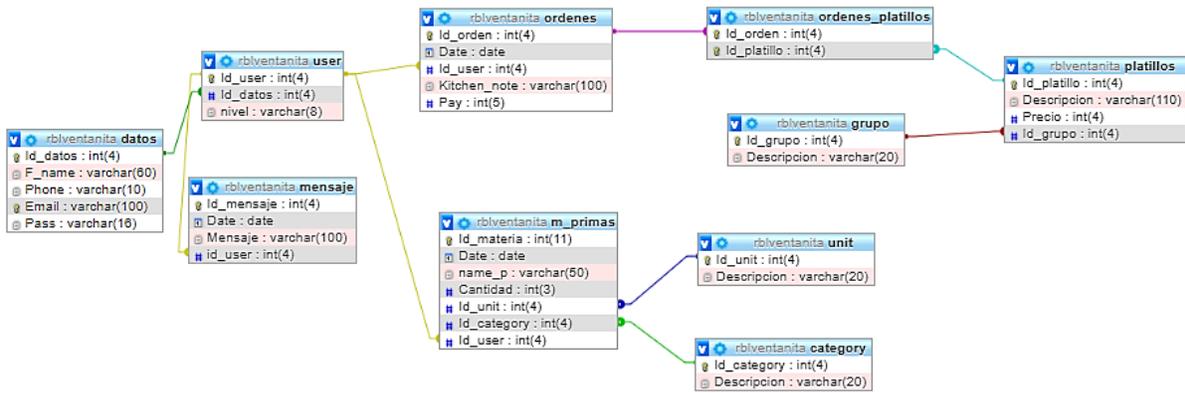


Figura 3. Diagrama de Base de datos

Tablas de diccionario de datos de la base de datos

Para una mejor comprensión de la base de datos se realizó el diccionario de datos para las entidades: datos (**Tabla 1.** Diccionario de datos de la tabla “datos”), user (**Tabla 2.** Diccionario de datos de la tabla “user”), mensaje (**Tabla 3.** Diccionario de datos de la tabla “mensaje”), orden (**Tabla 4.** Diccionario de datos de la tabla “orden”), orden_ platillo (**Tabla 5.** Diccionario de datos de la tabla “orden_ platillo”), platillos (**Tabla 6.** Diccionario de datos de la tabla “platillos”), grupo (**Tabla 7.** Diccionario de datos de la tabla “grupo”), m_primas (**Tabla 8.** Diccionario de datos de la tabla “m_primas”), unidad (**Tabla 9.** Diccionario de datos de la tabla “unidad”) y categoría (**Tabla 10.** Diccionario de datos de la tabla “categoría”). Con lo anterior se buscó especificar la función de cada atributo de las entidades de la base de datos.

| datos | | | | | |
|---|----------|------------|--------|-----------|--|
| Descripción de tabla que almacena datos de los gerentes | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción |
| PK | Id_datos | Int | 4 | Si | Identificador único de los datos almacenados |
| | F_name | Varchar | 60 | Si | Nombre completo de la persona registrada |
| | Phone | Varchar | 10 | Si | Número de teléfono |
| | Email | Varchar | 100 | Si | Correo electrónico con el que ingresará al sistema |
| | Pass | Varchar | 16 | Si | Contraseña para entrar al sistema |

Tabla 1. Diccionario de datos de la tabla “datos”.

| user | | | | | | |
|---|----------|------------|--------|-----------|---|---|
| Descripción de tabla que almacena datos de los gerentes | | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción | |
| PK | Id_user | Int | 4 | Si | Identificador de usuario | o |
| FK | Id_datos | Int | 4 | Si | Identificador que asociara a un usuario con un registro en la tabla "datos" | o |
| | Nivel | Varchar | 8 | Si | Mesero o Gerente | |

Tabla 2. Diccionario de datos de la tabla "user".

| mensaje | | | | | | |
|---|------------|------------|--------|-----------|--|---|
| Descripción de tabla que almacena datos de los gerentes | | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción | |
| PK | Id_mensjae | Int | 4 | Si | Identificador de mensaje | |
| | Date | Date | | Si | Fecha de realización del mensaje | |
| | Mensaje | Varchar | 100 | Si | Mensaje en cuestión | |
| PK | Id_user | Int | 4 | Si | Identificador que nos ayudara a conocer quien realizo el mensaje | o |

Tabla 3. Diccionario de datos de la tabla "mensaje".

| ordenes | | | | | | |
|---|--------------|------------|--------|-----------|---|---|
| Descripción de tabla que almacena datos de los gerentes | | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción | |
| PK | Id_orden | Int | 5 | Si | Identificador único de la orden | o |
| FK | Id_user | Int | 4 | Si | Identificador que nos servirá para almacenar información de quien realizo la nota | o |
| | Date | Date | | Si | Fecha de la orden | |
| | Kitchen_note | Varchar | 100 | Si | Nota para cocina | |
| | Pay | Int | 5 | Si | Total de cuenta | |

Tabla 4. Diccionario de datos de la tabla "orden".

| ordenes_platillo | | | | | | |
|---|-------------|------------|--------|-----------|--|---|
| Descripción de tabla que almacena datos de los gerentes | | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción | |
| PK | Id_orden | Int | 4 | Si | Este apartado nos servirá para asociar varios platillos pedidos a una sola nota | o |
| PK | Id_platillo | int | 4 | Si | Identificador de que nos servirá para conocer que platillos pidió nuestro comensal | o |

Tabla 5. Diccionario de datos de la tabla "orden_platillo".

| platillos | | | | | |
|---|-------------|------------|--------|-----------|---|
| Descripción de tabla que almacena datos de los gerentes | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción |
| PK | Id_platillo | Int | 4 | Si | Identificador del platillo que nos servirá para pasar información a la tabla "ordenes_platillo" |
| FK | Id_grupo | Int | 4 | Si | Categoría en la que se encuentra nuestro platillo |
| | Descripción | Varchar | 110 | Si | Nombre y descripción del producto |
| | Precio | Int | 5 | Si | Valor del producto |

Tabla 6. Diccionario de datos de la tabla "platillos".

| grupo | | | | | |
|---|-------------|------------|--------|-----------|------------------------------------|
| Descripción de tabla que almacena datos de los gerentes | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción |
| PK | Id_grupo | Int | 4 | Si | Identificador de nuestra categoría |
| | Descripción | Varchar | 20 | Si | Nombre de la categoría |

Tabla 7. Diccionario de datos de la tabla "grupo".

| m_primas | | | | | |
|---|-------------|------------|--------|-----------|--|
| Descripción de tabla que almacena datos de los gerentes | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción |
| PK | Id_materia | Int | 4 | Si | Identificador de la materia prima registrada |
| FK | Id_user | Int | 4 | Si | Identificador del usuario que hizo el registro |
| | Date | Date | | Si | Fecha de registro o modificación |
| FK | Id_category | Int | 4 | Si | Identificador de la categoría a la que pertenece el producto |
| | Name_p | Varchar | 50 | Si | Nombre del producto |
| | Cantidad | Int | 3 | Si | Cantidad en almacén |
| FK | Id_unit | Int | 4 | Si | Identificador del tipo de unidad de medición del producto |

Tabla 8. Diccionario de datos de la tabla "m_primas".

| unit | | | | | |
|---|-------------|------------|--------|-----------|--------------------------------------|
| Descripción de tabla que almacena datos de los gerentes | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción |
| PK | Id_unit | Int | 4 | Si | Identificador de la unidad de medida |
| | Descripción | Varchar | 20 | Si | Nombre de la unidad de medición |

Tabla 9. Diccionario de datos de la tabla "unidad".

| category | | | | | |
|---|-------------|------------|--------|-----------|-------------------------------|
| Descripción de tabla que almacena datos de los gerentes | | | | | |
| Clave | Campo | T. de dato | Tamaño | Requerido | Descripción |
| PK | Id_category | Int | 4 | Si | Identificador de la categoría |
| | Descripción | Varchar | 20 | Si | Nombre de la categoría |

Tabla 10. Diccionario de datos de la tabla "categoría".

Interfaces

Para el diseño de los primeros prototipos se eligió los de tipo “baja fidelidad”, para posteriormente (sección 3.3 Construcción de prototipo) pasar a un híbrido de los tipos prototipo “modular” y prototipo de “Alta fidelidad”. Se eligieron de esta manera pues se consideró el poco tiempo que se tuvo para trabajar y que se tenía que mostrar resultados rápidamente.

El sistema se diseñó de manera que sea fácil de manipular buscando que los usuarios se familiaricen con él rápidamente.

Login

El sistema se diseñó de forma que 2 tipos de usuarios (meseros y gerentes) puedan acceder y se elaboró dos dibujos para después pasar a la construcción del prototipo (figura 4 y 5).

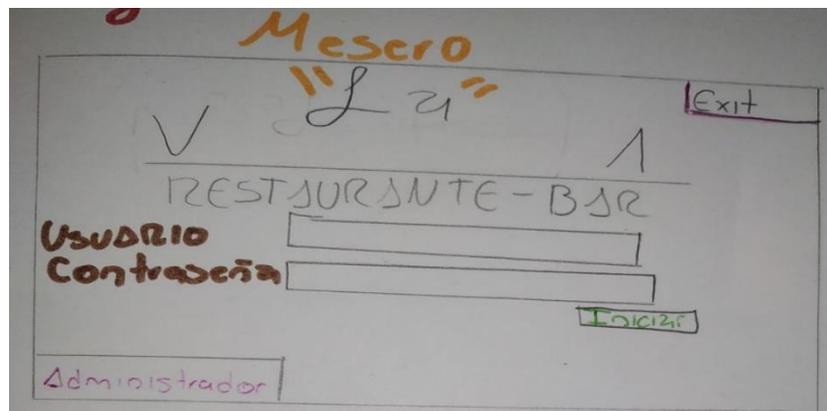


Figura 4. Diseño Login Mesero.

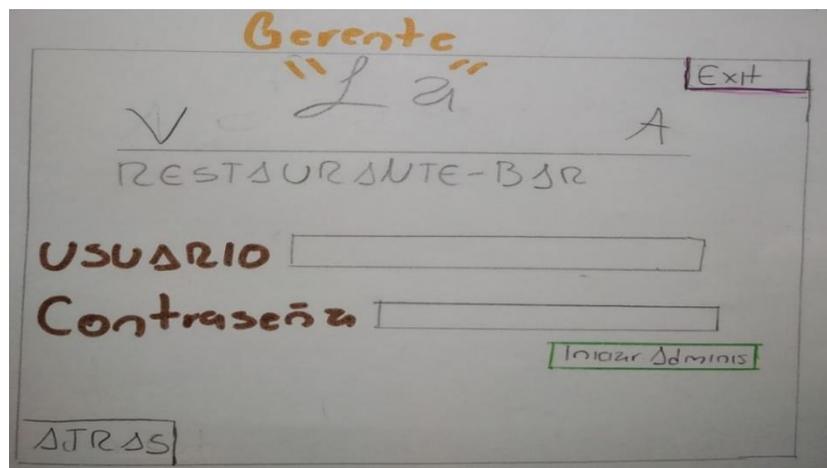


Figura 5. Diseño Login Gerente.

Menús Gerente y Mesero

Al contar con 2 tipos de usuario, es importante considerar que cada uno desempeñara una tarea distinta en el sistema, por eso existen 2 menús, uno con todas las opciones (**Figura 6**. Diseño menú gerente) y otro con opciones reducidas (**Figura 7**. Diseño menú gerente).

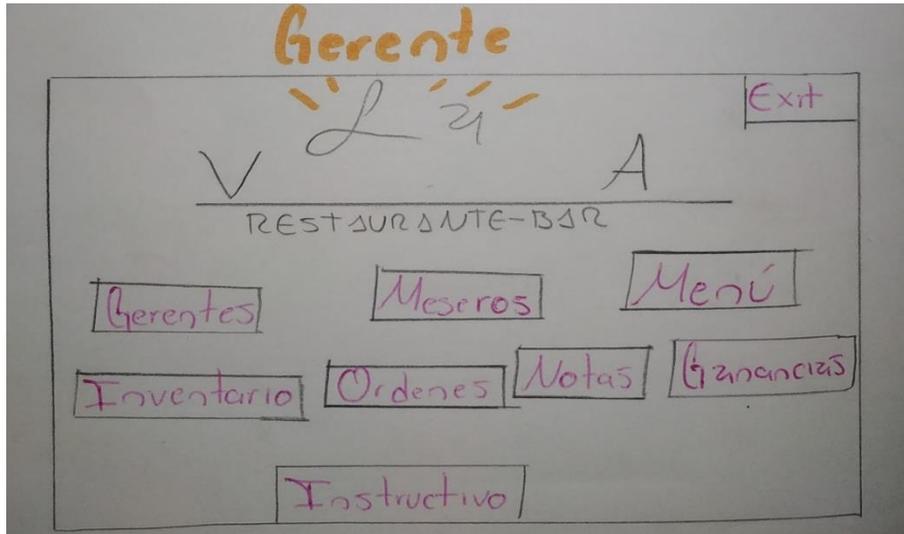


Figura 6. Diseño menú gerente.

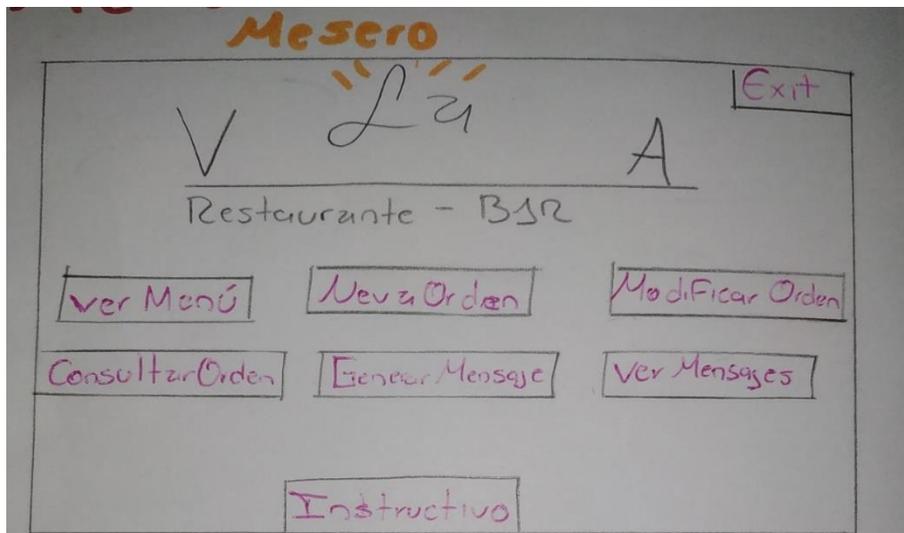


Figura 7. Diseño menú mesero.

Menús de cada opción para gerente

Los diseños para los menús de gerentes, meseros, inventario, menú (o carta) y órdenes se diseñaron de forma que tuvieran opciones similares, las cuales son

agregar, consultar, modificar, eliminar y registros, por eso se optó por un diseño muy similar para estas interfaces (**Figura 8**. Diseño de menú para gerente, meseros, inventario, menús (o carta) y ordenes).

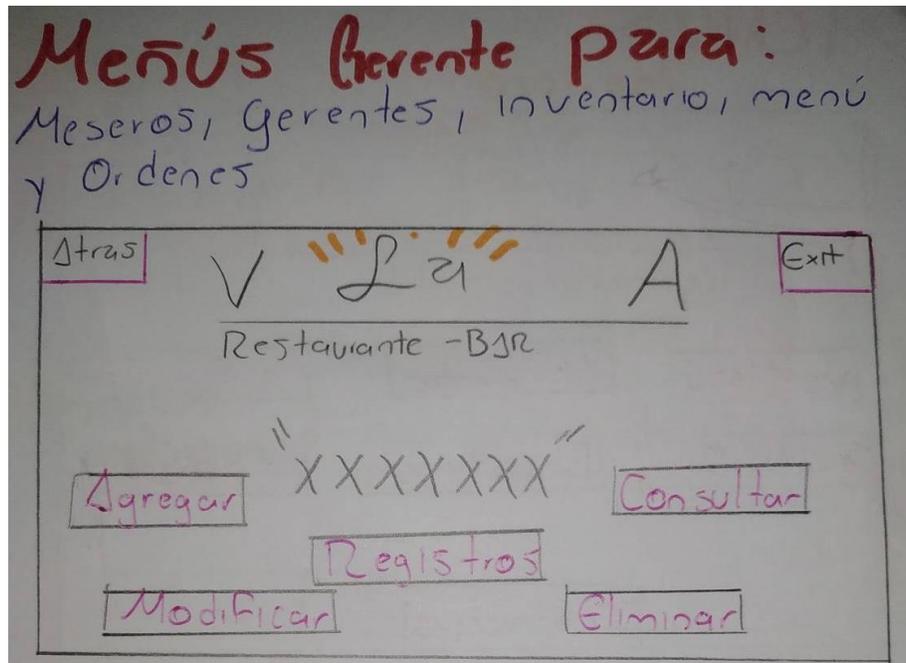


Figura 8. Diseño de menú para gerente, meseros, inventario, menús (o carta) y órdenes.

Para los menús de Ganancias y Notas se optó por un diseño diferente pues se consideró las diferentes opciones que tienen cada uno. Originalmente la interfaz de ganancias (**Figura 9**. Diseño interfaz ganancias) contaba con 3 opciones principales las cuales son: generar el registro del día, consultar ganancias de un día específico y ver los registros de todos los días almacenados y para la interfaz de notas (**Figura 10**. Diseño interfaz notas) sólo se consideraron 2 opciones (generar y ver todas las notas).

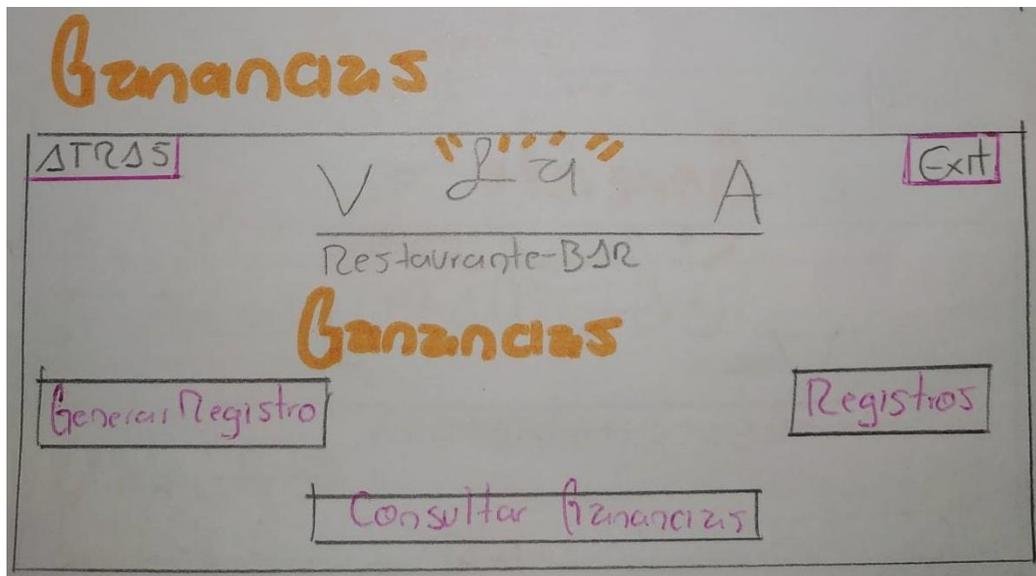


Figura 9. Diseño interfaz ganancias.

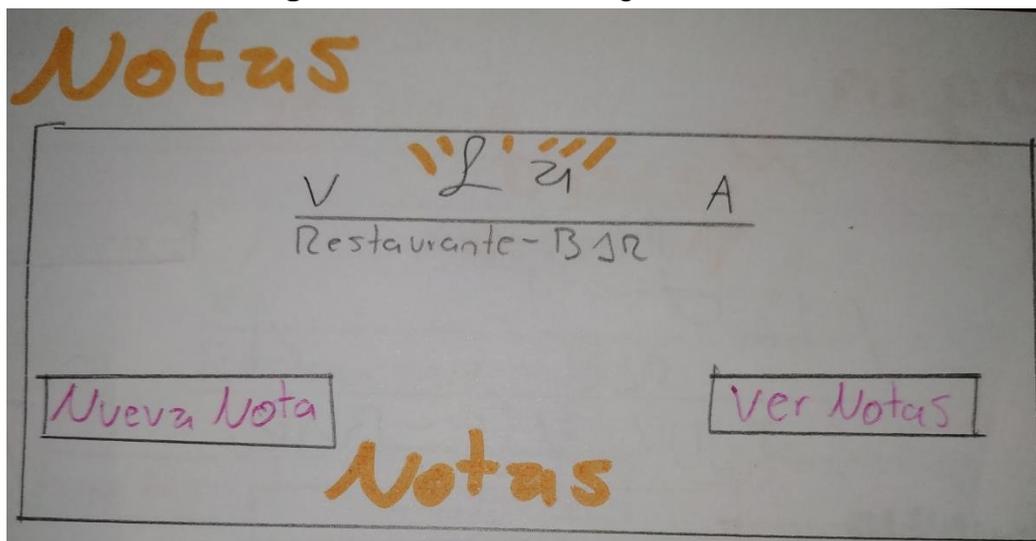


Figura 10. Diseño interfaz notas.

Para las demás interfaces no se realizó un diseño de prototipo de tipo “baja fidelidad”, pues se pasó directamente al prototipo tipo híbrido entre “alta fidelidad y modular”

3.3 Construcción del prototipo

Para esta etapa dejamos atrás el diseño y pasamos a la implementación de la base de datos e interfaces.

Script de la base de datos

Para el diseño y desarrollo de la base de datos se hizo uso de la herramienta de phpMyAdmin.

```
CREATE TABLE `category` ( `Id_category` int(4) NOT NULL, `Descripcion`
varchar(20) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
CREATE TABLE `datos` ( `Id_datos` int(4) NOT NULL, `F_name` varchar(60)
NOT NULL, `Phone` varchar(10) NOT NULL, `Email` varchar(100) NOT NULL,
`Pass` varchar(16) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
CREATE TABLE `grupo` ( `Id_grupo` int(4) NOT NULL, `Descripcion` varchar(20)
NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
CREATE TABLE `mensaje` ( `Id_mensaje` int(4) NOT NULL, `Date` date NOT
NULL, `Mensaje` varchar(100) NOT NULL, `id_user` int(4) NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
CREATE TABLE `m_primas` ( `Id_materia` int(11) NOT NULL, `Date` date NOT
NULL, `name_p` varchar(50) NOT NULL, `Cantidad` int(3) NOT NULL, `Id_unit`
int(4) NOT NULL, `Id_category` int(4) NOT NULL, `Id_user` int(4) NOT NULL)
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
CREATE TABLE `ordenes` ( `Id_orden` int(4) NOT NULL, `Date` date NOT
NULL, `Id_user` int(4) NOT NULL, `Kitchen_note` varchar(100) NOT NULL,
`Pay` int(5) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
CREATE TABLE `ordenes_platillos` ( `Id_orden` int(4) NOT NULL, `Id_platillo`
int(4) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
--
```

```
CREATE TABLE `platillos` ( `Id_platillo` int(4) NOT NULL, `Descripcion`  
varchar(110) NOT NULL, `Precio` int(4) NOT NULL, `Id_grupo` int(4) NOT NULL)  
ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
```

```
CREATE TABLE `unit` ( `Id_unit` int(4) NOT NULL, `Descripcion` varchar(20)  
NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
--
```

```
CREATE TABLE `user` ( `Id_user` int(4) NOT NULL, `Id_datos` int(4) NOT NULL,  
`nivel` varchar(8) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
-- Índices para tablas volcadas
```

```
-- Indices de la tabla `category`
```

```
ALTER TABLE `category`  
ADD PRIMARY KEY (`Id_category`);
```

```
-- Indices de la tabla `datos`
```

```
ALTER TABLE `datos`  
ADD PRIMARY KEY (`Id_datos`), ADD UNIQUE KEY `Id_datos` (`Id_datos`),  
ADD UNIQUE KEY `Email` (`Email`);
```

```
-- Indices de la tabla `grupo`
```

```
ALTER TABLE `grupo`  
ADD PRIMARY KEY (`Id_grupo`);
```

```
-- Indices de la tabla `mensaje`
```

```
ALTER TABLE `mensaje`  
ADD PRIMARY KEY (`Id_mensaje`), ADD UNIQUE KEY `Id_mensaje`  
(`Id_mensaje`), ADD KEY `id_user` (`id_user`);
```

```
-- Indices de la tabla `m_primas`
```

```
ALTER TABLE `m_primas`  
ADD PRIMARY KEY (`Id_materia`), ADD UNIQUE KEY `Id_materia`  
(`Id_materia`), ADD KEY `Id_category` (`Id_category`), ADD KEY `Id_unit`  
(`Id_unit`), ADD KEY `Id_user` (`Id_user`);
```

```

-- Indices de la tabla `ordenes`
ALTER TABLE `ordenes`
  ADD PRIMARY KEY (`Id_orden`), ADD UNIQUE KEY `Id_orden` (`Id_orden`),
  ADD KEY `Id_user` (`Id_user`);
-- Indices de la tabla `ordenes_platillos`
ALTER TABLE `ordenes_platillos`
  ADD PRIMARY KEY (`Id_orden`,`Id_platillo`), ADD KEY `Id_platillo`
  (`Id_platillo`);
-- Indices de la tabla `platillos`
ALTER TABLE `platillos`
  ADD PRIMARY KEY (`Id_platillo`), ADD UNIQUE KEY `Id_platillo` (`Id_platillo`),
  ADD KEY `Id_grupo` (`Id_grupo`);
-- Indices de la tabla `unit`
ALTER TABLE `unit`
  ADD PRIMARY KEY (`Id_unit`);
-- Indices de la tabla `user`
ALTER TABLE `user`
  ADD PRIMARY KEY (`Id_user`), ADD UNIQUE KEY `Id_user` (`Id_user`), ADD
  UNIQUE KEY, ADD KEY `Id_datos` (`Id_datos`);

-- AUTO_INCREMENT de las tablas volcadas

-- AUTO_INCREMENT de la tabla `category`
ALTER TABLE `category`
  MODIFY `Id_category` int(4) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `datos`
ALTER TABLE `datos`
  MODIFY `Id_datos` int(4) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `grupo`
ALTER TABLE `grupo`
  MODIFY `Id_grupo` int(4) NOT NULL AUTO_INCREMENT;

```

```

-- AUTO_INCREMENT de la tabla `mensaje`
ALTER TABLE `mensaje`
  MODIFY `Id_mensaje` int(4) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `m_primas`
ALTER TABLE `m_primas`
  MODIFY `Id_materia` int(11) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `ordenes`
ALTER TABLE `ordenes`
  MODIFY `Id_orden` int(4) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `platillos`
ALTER TABLE `platillos`
  MODIFY `Id_platillo` int(4) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `unit`
ALTER TABLE `unit`
  MODIFY `Id_unit` int(4) NOT NULL AUTO_INCREMENT;
-- AUTO_INCREMENT de la tabla `user`
ALTER TABLE `user`
  MODIFY `Id_user` int(4) NOT NULL AUTO_INCREMENT;

-- Restricciones para tablas volcadas

-- Filtros para la tabla `mensaje`
ALTER TABLE `mensaje`
  ADD CONSTRAINT `mensaje_ibfk_1` FOREIGN KEY (`id_user`) REFERENCES
`user` (`Id_user`);
-- Filtros para la tabla `m_primas`
ALTER TABLE `m_primas`
  ADD CONSTRAINT `m_primas_ibfk_1` FOREIGN KEY (`Id_category`)
REFERENCES `category` (`Id_category`),
  ADD CONSTRAINT `m_primas_ibfk_2` FOREIGN KEY (`Id_unit`)
REFERENCES `unit` (`Id_unit`),

```

```

    ADD CONSTRAINT `m_primas_ibfk_3` FOREIGN KEY (`Id_user`)
REFERENCES `user` (`Id_user`);
-- Filtros para la tabla `ordenes`
ALTER TABLE `ordenes`
    ADD CONSTRAINT `ordenes_ibfk_1` FOREIGN KEY (`Id_user`) REFERENCES
`user` (`Id_user`);
-- Filtros para la tabla `ordenes_platillos`
ALTER TABLE `ordenes_platillos`
    ADD CONSTRAINT `ordenes_platillos_ibfk_1` FOREIGN KEY (`Id_orden`)
REFERENCES `ordenes` (`Id_orden`),
    ADD CONSTRAINT `ordenes_platillos_ibfk_2` FOREIGN KEY (`Id_platillo`)
REFERENCES `platillos` (`Id_platillo`);
-- Filtros para la tabla `platillos`
ALTER TABLE `platillos`
    ADD CONSTRAINT `platillos_ibfk_1` FOREIGN KEY (`Id_grupo`) REFERENCES
`grupo` (`Id_grupo`);
-- Filtros para la tabla `user`
ALTER TABLE `user`
    ADD CONSTRAINT `user_ibfk_1` FOREIGN KEY (`Id_datos`) REFERENCES
`datos` (`Id_datos`);
COMMIT;

```

Interfaces graficas

El desarrollo de las interfaces empezó de afuera hacia adentro, teniendo como base los accesos para los 2 tipos de usuarios (meseros y gerentes), menús principales (los que sólo cuentan con botones que nos re-direccionan a otras interfaces) (**Figura 11**. Fondo para login y menús) y formularios (donde están las funciones de manipular datos de la base de datos) (**Figura 12**. Fondo para formularios) tengan como tamaño 800 x 500 pixeles, teniendo como fondo 2 diferentes imágenes que se realizaron haciendo uso del editor de imágenes de Windows (Paint).

Para la construcción de los prototipos pasamos del papel a propiamente el diseño que tendrán al final las interfaces, se hizo uso de las herramientas que tiene Netbeans para diseño de las mismas.



Figura 11. Fondo para login y menús.

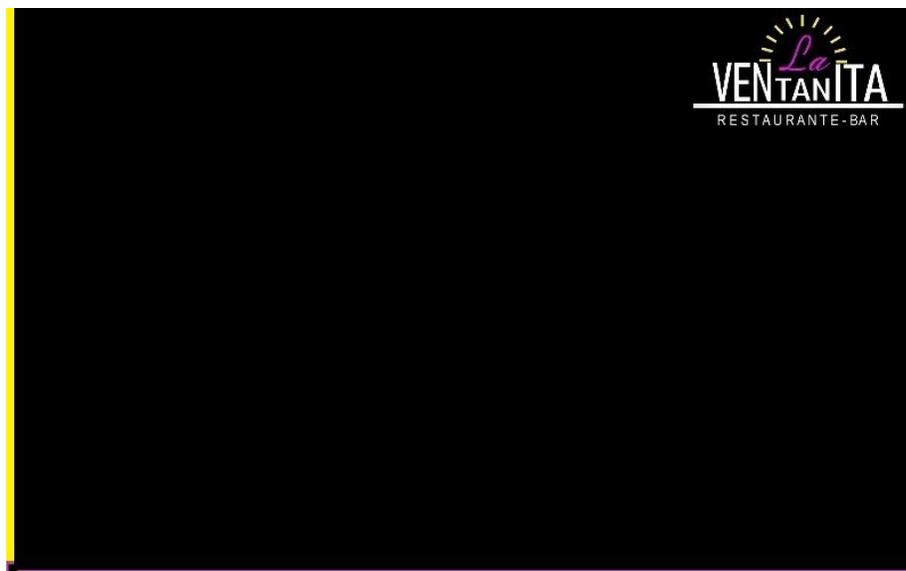


Figura 12. Fondo para formularios.

Se utilizó el fondo negro a petición del gerente para que el sistema tenga armonía de color con la carta que se maneja en el restaurante.

Todas las interfaces se implementaron buscando sencillez para que los usuarios se adapten fácilmente al uso del sistema.

Interfaces de Gerente

Se empezó a diseñar de fuera hacia adentro, empezando por la interfaz de login (Figura 13. Interfaz login para Gerentes) que cuenta con 3 botones principales (salir, atrás e iniciar) y 2 campos de formulario. Los botes sirven para salir, regresar a la interfaz de Login de meseros (Figura 14. Interfaz login para meseros) e iniciar sesión. El botón de “iniciar ADMINISTRADOR” nos re-direcciona al menú de gerente (Figura 15. Interfaz menú gerente). En la interfaz de menú se observan varias opciones (gerentes, meseros, menú (o carta), inventario, órdenes, notas y ganancias) y cada una de ellas nos lleva a un menú distinto dependiendo del apartado. Las opciones dentro de los menús internos para meseros, gerentes, inventario, menú (o carta) y ordenes son similares (agregar, modificar, consultar, eliminar y ver registros) (Figura 16 a 20). Para los menús de notas y ganancias se cuenta con 2 botones (Nueva nota para meseros) (Figura 21. Interfaz menú manipulación notas) y 3 botones (Generar registro del día, registros y consultar ganancias) (Figura 22. Interfaz menú manipulación ganancias) respectivamente.



Figura 13. Interfaz login para gerentes.



Figura 14. Interfaz login para meseros.



Figura 15. Interfaz menú gerente.



Figura 16. Interfaz menú manipulación gerente.



Figura 17. Interfaz menú manipulación mesero.



Figura 18. Interfaz menú manipulación menú.



Figura 19. Interfaz menú manipulación inventario.



Figura 20. Interfaz menú manipulación órdenes.



Figura 21. Interfaz menú manipulación notas.



Figura 22. Interfaz menú manipulación ganancias.

Interfaces de Mesero

Al igual que el gerente, el diseño de interfaces para mesero empezó de afuera hacia adentro. Es importante mencionar que el sistema inicia automáticamente con la interfaz de login mesero (**Figura 14.** Interfaz login para meseros) que cuenta con 3 botones (salir, iniciar y administrador) y 2 formularios (usuario y contraseña), el botón de administrador sirve para re-direccionarnos a la interfaz de login gerente (**Figura 15.** Interfaz login para gerentes). El botón más relevante en la interfaz de login mesero es el de iniciar, que es el que nos re-direcciona al menú para meseros (**Figura 23.** Interfaz menú mesero); en el menú mesero cuenta con 8 botones funcionales (salir, ver menú, nueva orden, modificar orden, consultar ordenes, mensaje para gerentes(generar), mensajes para meseros (consultar), leerme antes de empezar a manejar el sistema). Estas son las únicas funciones que el mesero puede manipular.



Figura 23. Interfaz menú mesero

Formularios

La mayoría de los formularios se diseñaron sólo para que un usuario de tipo gerente pueda manipularlos.

Se describirá el funcionamiento de las interfaces de los formularios que tiene un usuario gerente para la manipulación de datos de meseros. Para la manipulación de datos de los meseros, se cuenta con 5 opciones:

- Agregar (**Figura 24.** Interfaz agregar nuevo mesero), sirve para guardar los datos de un nuevo usuario tipo mesero, y cuenta con 6 apartados editables (nombre(s), apellido paterno, apellido materno, número de teléfono, usuario y contraseña) y 1 que el sistema genera automáticamente (El identificador).
- Eliminar (**Figura 25.** Interfaz eliminar mesero), sirve para que en su caso se borre el registro de un mesero, la interfaz cuenta con un botón que sirve para buscar mediante el identificador del mesero y rellenar los campos de datos, una vez que se selecciona el mesero en cuestión a eliminar se procede a presionar el botón de eliminar para borrar el registro.
- Consultar (**Figura 26.** Interfaz consultar datos de un mesero), sirve para consultar la información de un mesero mediante el uso de un botón de búsqueda

de Identificador. En esta interfaz se mostrarán los datos almacenados de un mesero específico.

- Modificar (**Figura 27.** Interfaz modificar a un mesero), al igual que en la interfaz de eliminar se cuenta con un botón que sirve para buscar mediante el identificador del mesero en la base de datos a un mesero del cual se desea modificar su información. Es importante mencionar que en esta interfaz de modificación se diseñó de manera que sea posible manipular todos sus datos excepto el apartado de Identificador por cuestiones de seguridad.
- Registros (**Figura 28.** Interfaz registros de un mesero), en esta interfaz existe una tabla que servirá para ver todos los registros de la base de datos referente a meseros, es meramente para conocer todos los datos almacenados.

The screenshot shows a web interface titled "Nuevo Mesero" for "La VENTANITA RESTAURANTE - BAR". It features a form with the following fields and labels:

- Nombre(s):** Nombre o nombres del nuevo MESERO (35)
- Apellido Paterno:** Apellido Paterno (35)
- Apellido Materno:** Apellido Materno (35)
- Telefono:** Numero de telefono a 10 digitos
- Usuario:** Usuario con el que se ingresara al sistema
- Contraseña:** Contraseña

Additional elements include a "Limpiar campos" button below the password field, an "Alta Mesero :)" button to the right, and a "Menu 'Meseros'" button at the bottom left. A search box labeled "Id_mesero" is visible at the top right.

Figura 24. Interfaz agregar nuevo mesero

Borrar Mesero

La VENTANITA RESTAURANTE-BAR

ID Mesero: **Item 1**

Nombre(s)

Apellido Paterno

Apellido Materno

Telefono

Usuario

Contraseña

Limpiar campos

Eliminar :(

Menu "Meseros"

Figura 25. Interfaz eliminar a un mesero

Ver datos de un Mesero

La VENTANITA RESTAURANTE-BAR

ID Mesero: **Item 1**

Nombre(s)

Apellido Paterno

Apellido Materno

Telefono

Usuario

Contraseña

Limpiar campos

Menu "Meseros"

Figura 26. Interfaz consultar a un mesero

Modificar datos de un Mesero

La VENTANITA RESTAURANTE-BAR

ID Mesero: Item 1

Nombre(s):

Apellido Paterno:

Apellido Materno:

Telefono:

Usuario:

Contraseña:

Limpiar campos

Modificar

Menu "Meseros"

Figura 27. Interfaz modificar a un mesero

Registros de Meseros

La VENTANITA RESTAURANTE-BAR

| Title 1 | Title 2 | Title 3 | Title 4 | Title 5 | Title 6 |
|---------|---------|---------|---------|---------|---------|
| | | | | | |
| | | | | | |
| | | | | | |

Menu "Meseros"

Figura 28. Interfaz registros de meseros

3.4 Desarrollo y evaluación del prototipo por el cliente

Después del diseño, desarrollo de la base de datos y las interfaces gráficas se pasó a la evaluación por parte del cliente. La evaluación del primer prototipo del software (que aún no contaba con conexión a la base de datos) se llevó a cabo mediante una entrevista que se le realizó al gerente del restaurante después de que navegara entre las interfaces del programa.

Después de la entrevista y posteriormente escuchar los comentarios del gerente se llegó a varias conclusiones.

Conclusiones obtenidas referentes al prototipo mostrado

1. Interfaz menú gerente (**Figura 15**. Interfaz menú gerente).
 - El gerente no genera órdenes, sólo cobra las ya existentes y ve el contenido de las mismas.
 - El apartado notas debe llamarse “mensajes para meseros”.
 - Unir en uno sólo los apartados de gerentes y meseros llamándolo “usuarios”
 - Quitar el botón de “leerme antes de empezar a manejar el sistema”
2. Interfaz de manipulación de gerentes y menú (**Figura 16 y 17**).
 - Para el nuevo diseño quitar el botón “eliminar” e incluir esa opción el apartado de consultar (para todas las demás interfaces igual).
 - Que el formulario para agregar datos sea uno sólo sin darle un rol hasta ese punto al nuevo registro.
 - Que exista un apartado en donde se pueda asociar un rol a un registro existente.
 - Agregar un botón que nos lleve a una interfaz en donde se pueda seleccionar el id de un usuario y nos muestre su nombre, los datos asociados a él, así como su rol en el sistema.
 - Agregar un apartado que se llame empleados para poder ver todos los empleados con un rol específico.
3. Interfaz de manipulación de menú (**Figura 18**. Interfaz menú manipulación menú.)

- Eliminar el apartado de “eliminar artículo”.
 - Agregar 4 opciones para “Grupo” (nuevo grupo, modificar datos, registro, datos de un grupo), esto para poder agregar categorías que posteriormente se servirán para la consulta de platillos.
4. Interfaz de manipulación de inventario (**Figura 19**. Interfaz menú manipulación inventario.)
- Agregar 4 opciones para “unidades” (nueva unidad, modificar datos, registro, ver unidades y registro) esto para poder agregar unidades que posteriormente se agregarán a los productos de inventario.
 - Lo mismo, pero en un nuevo apartado de “categorías”.
5. Interfaz de Manipulación de órdenes (**Figura 20**. Interfaz menú manipulación de órdenes.)
- Quitar todos los botones.
 - Agregar 3 botones que se llamen “calcular total”, “registros” (se podrán consultar todos los productos que están asociados a una orden) y cobrar cuenta.
6. Interfaz de Manipulación de notas (**Figura 21**. Interfaz menú manipulación de notas.)
- Cambiar el nombre a “mensajes”.
 - Cambiar el nombre del botón registro a “mensajes por fecha”.
 - Cambiar el nombre del botón nueva nota para meseros a “nuevo mensaje para meseros”.
7. Interfaz “menú ganancias (**Figura 22**. Interfaz menú manipulación de ganancias).
- Borrar todos los botones y sólo dejar uno en donde se pueda seleccionar el día del cual se desea calcular las ganancias.
8. Interfaz menú “mesero” (**Figura 23**. Interfaz menú mesero).
- Eliminar el botón de “Leerme antes de empezar a manejar el sistema”.
 - Cambiar el nombre del botón nueva orden a “nuevo registro de orden”.
 - Agregar un botón con el nombre de “Agregar un producto a una orden”.
 - Eliminar el botón de mensaje para gerente.

- Eliminar el botón de “modificar orden”.
9. Cambiar todas las interfaces que incluyan formularios adaptándolas a las modificaciones sugeridas.

3.5 Refinamiento del prototipo

Tomando en cuenta los comentarios del gerente se realizaron los cambios pertinentes a las interfaces y formularios. A continuación, muestro el prototipo final de todas las interfaces con las que cuenta el software.

El resultado final de la modificación al menú principal para los usuarios de tipo gerente se puede observar en la **figura 29**.



Figura 29. Nueva interfaz “menú para gerente”.

El resultado de la unificación que sufrieron los apartados de “meseros y gerentes” se observa en la **figura 30**, en donde además de mostrar los menús principales se observan todos los formularios en los que puede interactuar nuestro usuario.



Figura 30. Nuevas interfaces para el apartado de “Usuarios”.

Para la sección de mensajes se consideró el cambio de nombre de la interfaz; el resultado puede ser observado en la figura 31, así como sus secciones manipulables.



Figura 31. Nuevas interfaces para el apartado de “Mensajes para meseros”.

El apartado de “Ganancias” sufrió un cambio significativo que puede ser observado en la **figura 32**, pues ahora a diferencia de su antiguo diseño (**figura 22**. Interfaz menú manipulación ganancias) únicamente se cuenta con un botón con la función de mostrar todas las fechas en las que se ha agregado un nuevo registro de orden.



Figura 32. Nueva interfaz para el apartado de “Ganancias”.

Para el apartado de órdenes se tomó en cuenta las actividades de las que se encarga el gerente, las cuales se resumen en el cobro y visualización de los platillos

que se asocian a las órdenes. A continuación, se muestra el nuevo diseño de las interfaces, así como todos sus campos manipulables. El resultado se puede observar en la **figura 33**.



Figura 33. Nuevas interfaces para el apartado de “Órdenes”.

Para la sección de “menú” (o carta) además de los 4 botones principales en el apartado de “platillos” se agregaron otros 4 botones con las mismas funcionalidades (agregar, consultar, modificar y listado) para un apartado llamado “grupos”.

En el apartado de “grupos” se manipula la información de los grupos a los que pertenecen los platillos (pizza, pastas, postres, etc.).

Todas las pantallas y formularios de manipulación del apartado “menú” se muestran en la **figura 34**.



Figura 34. Nuevas interfaces para el apartado de "Menú".

El menú de “inventario” (**figura 35**), es el que cuenta con más botones de interacción. Dentro de esta interfaz se encuentran los botones de manipulación para las secciones de unidades, inventario y categorías.



Figura 35. Nueva interfaz del menú “Unidades - inventario - categorías”.

Las opciones del apartado que tiene el menú de “inventario” son para la manipulación de:

- “Unidades” de medida: Los apartados que tiene la sección de “unidades” (**figura 36**) nos ayudan a agregar o modificar datos de las medidas con las que se cuantifican los productos del inventario. Es importante mencionar que las unidades almacenadas no necesariamente tienen que ser convencionales.
- “Categorías” de productos: Las opciones de este apartado (**figura 37**) nos ayudan a agregar o modificar datos de las categorías a las que pertenecen los productos y de esta forma no tenerlos en una categoría global.
- “Inventario”: Apartado que nos brinda opciones para la manipulación de los datos de las materias primas que tenemos registradas (**figura 38**).



Figura 36. Opciones de manipulación de la sección “Unidades”.

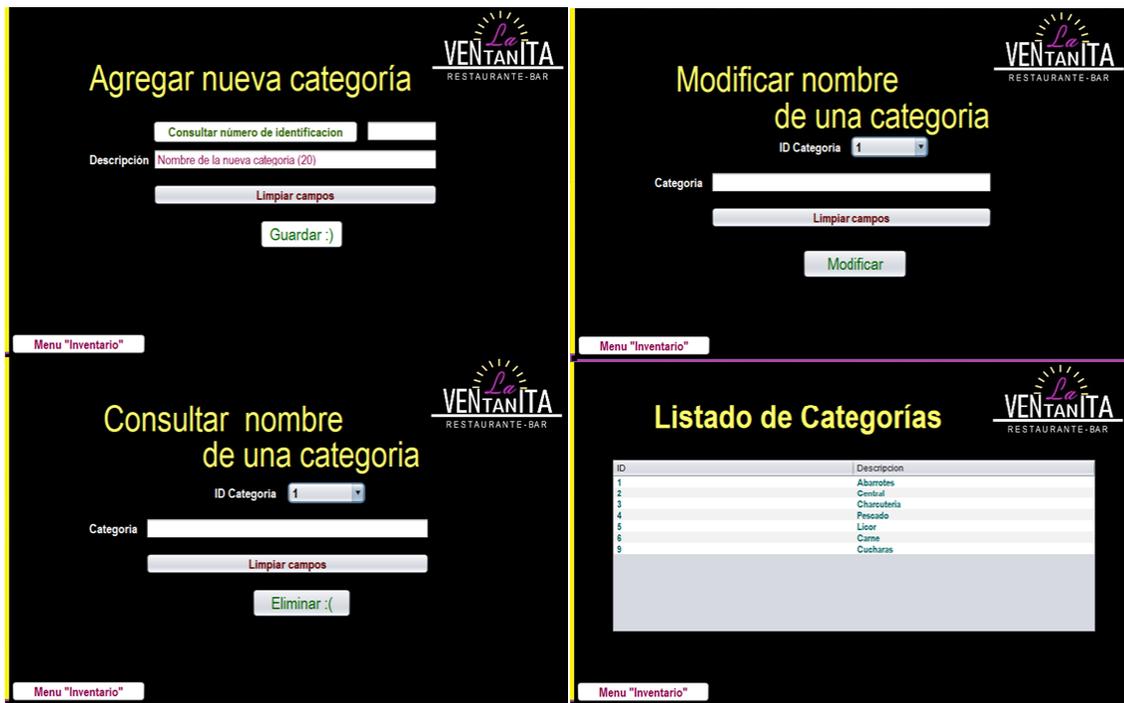


Figura 37. Opciones de manipulación de la sección “Categorías”.



Figura 38. Opciones de manipulación de la sección “Inventario”.

Se rediseñó el menú de interacción de los usuarios tipo “mesero” (figura 39) pensando en las actividades que desempeñan dentro del restaurante.



Figura 39. Nueva interfaz “Menú mesero”.

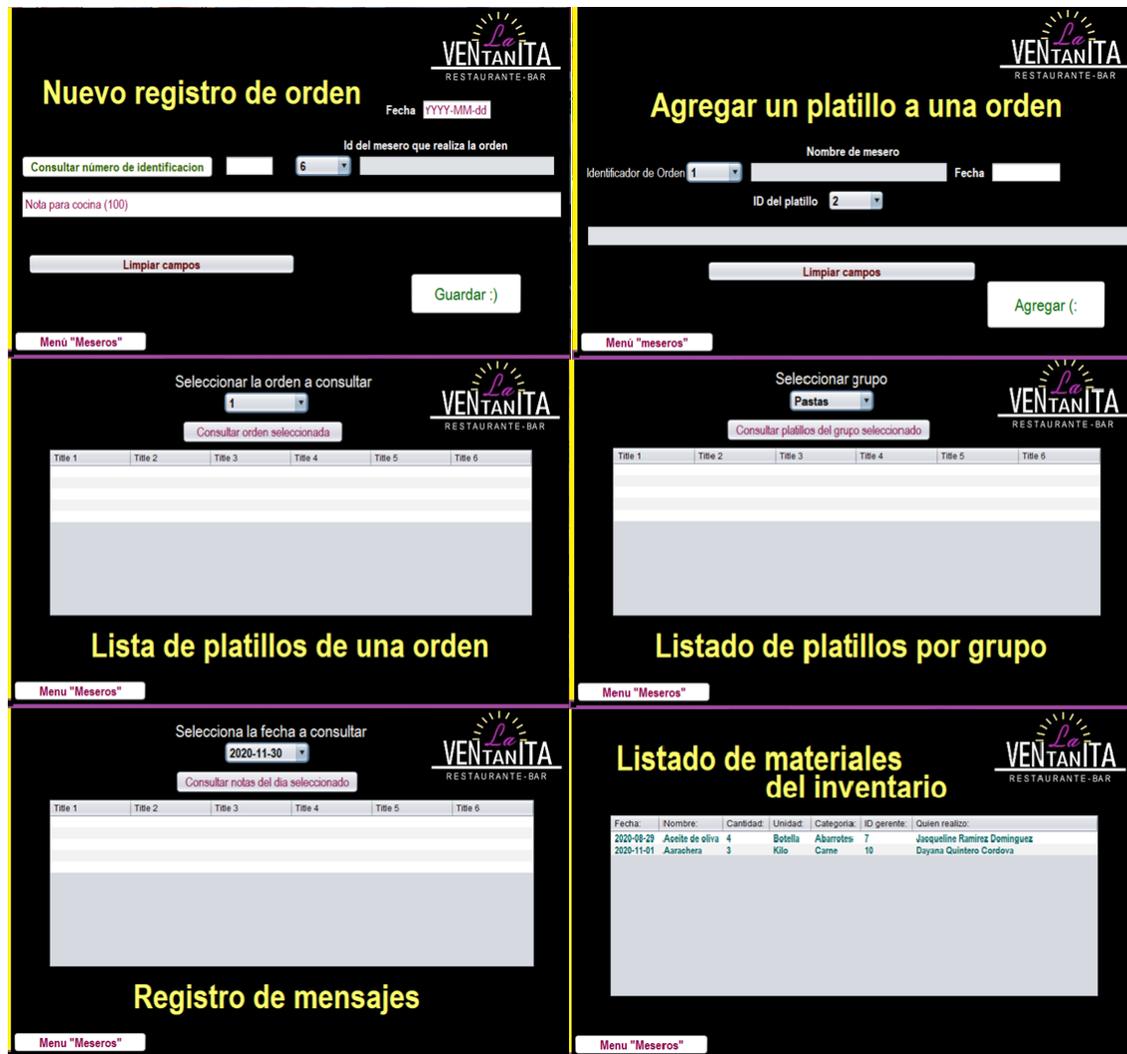


Figura 40. Opciones de manipulación para los usuarios de tipo “Mesero”.

Funcionalidad del sistema

Después del refinamiento de las interfaces, se procedió a darle funcionalidad a los apartados de formularios y consultas que hace el sistema, a continuación, se explicarán las funciones más relevantes.

Un problema que se detectó en un inicio fue que al agregar un nuevo registro este no mostraba que número de identificación tendría, para esto se agregó un botón con la función de hacer una consulta y el resultado pasarlo a un espacio de texto, de manera que el usuario pueda conocer qué número de identificación tendrá.

```

private void CNIJBActionPerformed(java.awt.event.ActionEvent evt) {
    Connection con = null;
    try {
        con = getConnection();
        ps = con.prepareStatement("SELECT * FROM user ORDER BY id_user DESC LIMIT 1");
        rs = ps.executeQuery();
        rs.next();
        CNIJTFC.setText(String.valueOf(rs.getInt("id_user") + 1));
    } catch (Exception e) {
        System.err.println(e);
    }
}

```

Figura 41. Consulta para traer número de identificación”

En la **figura 41** se observa la consulta que se ocupó la cual busca el último registro del sistema y lo devuelve sumándole 1 al resultado.

Para el apartado de consulta de “ganancias de un día” se ocupó un botón que muestra las fechas de cuando se hicieron registros de órdenes. Al seleccionar una fecha, se calcula automáticamente las ganancias del día seleccionado.

```

Connection s = null;
Connection con1 = null;
try {
    con1 = getConnection();
    ps = con1.prepareStatement("SELECT * FROM ordenes GROUP BY Date");
    rs = ps.executeQuery();
    while(rs.next())
    {
        ComboJCB.addItem(rs.getString("Date"));
    }
    contador++;
} catch (Exception e)
{
    System.out.println(e);
}

```

Figura 42. Consulta de fechas de órdenes.

Se hace una búsqueda de las fechas en donde se realizó un registro de orden, posteriormente se agrupan las fechas similares y se muestran en el botón antes mencionado. Para hacer la suma de ganancias, se hace uso de la función **SUM** de MySQL, la cual suma los valores de una columna de una tabla de la base de datos. Para hacer eso se unieron varias tablas en una con otra función de MySQL (INNER JOIN) (**figura 43**).

```

Connection con = null;
try {
    if(contador>0)
    {
        con = getConnection();
        ps = con.prepareStatement("SELECT SUM(ordenes.Pay) FROM 'ordenes' INNER JOIN 'ordenes_platillos' "
            + "on ordenes.Id_orden = ordenes_platillos.Id_orden INNER JOIN 'platillos' ON ordenes_platillos.Id_platillo "
            + " = platillos.Id_platillo WHERE ordenes.Date = '"+this.ComboJCB.getSelectedItem()+"'");
        rs = ps.executeQuery();
        rs.next();
        Ganancias.setText(String.valueOf(rs.getString("SUM(ordenes.Pay)")));
    }
}

```

Figura 43. Consulta para calcular las ganancias de un día seleccionado.

Para agregar un pedido, primero se tiene que seleccionar la orden a la que se le agregará el pedido y se mostrará el nombre del mesero que hizo el registro del pedido. Si nos fijamos en el diagrama de base de datos nos damos cuenta que los datos se almacenan en una tabla que se llama “datos”, y el pedido en la tabla “órdenes”. Para mandar a traer el dato del nombre del mesero se tuvo que unir la tabla órdenes, con la tabla user y la tabla datos (**figura 44**).

```

Connection con = null;
try {
    if(contador>0)
    {
        con = getConnection();
        ps = con.prepareStatement("SELECT ordenes.Date, datos.F_name FROM 'ordenes' INNER JOIN 'user' ON "
            + "' ordenes.Id_usuario = user.Id_usuario INNER JOIN 'datos' ON user.Id_datos = datos.Id_datos WHERE "
            + "' ordenes.Id_orden='"+this.Id_orden.getSelectedItem()+"'");
        rs = ps.executeQuery();
        rs.next();
        Datee.setText(String.valueOf(rs.getString("Date")));
        F_nameJTF.setText(String.valueOf(rs.getString("datos.F_name")));
    }
}

```

Figura 44. Unión de la tabla ordenes con user y datos.

Para la validación de los campos se hizo uso de la herramienta que nos ofrece NetBeans para la validación de letra por letra (KeyType), estas validaciones nos ayudan a que no haya conflictos al pasar datos del sistema a la base de datos, las validaciones más importantes son la de fecha (**figura 45**), correos electrónicos (**figura 46**), no poner letras en nombres de personas (**figura 47**), o poner letras en los campos en donde sólo se aceptan números (**figura 48**).

```

public boolean Em(String correo) {
    // Patrón para validar el email
    Pattern pattern = Pattern.compile("[_A-Za-z0-9-\\+]+(\\.[_A-Za-z0-9-\\+]*@)" + "[A-Za-z0-9-\\+](\\.[A-Za-z0-9-\\+]*)(\\.[A-Za-z]{2,})$");
    Matcher mather = pattern.matcher(correo);
    return mather.find();
}

```

Figura 45. Función que verifica que la estructura del correo electrónico se siga.

```

public boolean Fe(String Fecha)
{
    // Patrón para validar el email
    Pattern pattern = Pattern.compile("^\\d{4}-\\d{2}-\\d{2}$");
    Matcher mather = pattern.matcher(Fecha);
    return mather.find();
}

```

Figura 46. Función que verifica que la estructura de fechas que acepta MySQL se siga (AAAA-MM-dd).

```

int w = (int)evl.getKeyChar();
if ((w < 'a' || w > 'z') && (w < 'A' || w > 'Z') && (w < ' ' || w > ' ')) evl.consume();
if (Name_pJTFP.getText().length() == 50) evl.consume();

```

Figura 47. Validación para que sólo se ingresen letras, espacios y se respete el tamaño máximo permitido.

```

int w = (int)evl.getKeyChar();
if (w < '0' || w > '9') evl.consume();
if (CantidadJTFC.getText().length() == 3) evl.consume();

```

Figura 48. Validación para que sólo se ingresen números y se respete el tamaño máximo permitido.

Para mandar datos desde el software a la base de datos (INSERT, **figura 49**), primero se tienen que tomar los datos del campo de texto o de un botón, posteriormente pasarlo al tipo de dato que se especificó en la base de datos y mandarlo, para modificar (UPDATE, **figura 50**) es un proceso similar y para el

proceso de eliminar (DELETE, **figura 51**), el software sólo necesita conocer el identificador, posteriormente pasarlo al tipo de dato entero y hacer la modificación en la base de datos.

```
con = getConnection();
ps = con.prepareStatement("INSERT INTO user (Id_datos, Nivel) VALUES(?,?) ");
ps.setString(1, Id_datos.JTFC.getText());
ps.setString(2, (String) Nivel.JCB.getSelectedItem());
int resultado = ps.executeUpdate();
if(resultado > 0)
{
    JOptionPane.showMessageDialog(null, "asociacion exitosa");
    limpiarCajas();
}
else{
    JOptionPane.showMessageDialog(null, "Error al hacer la asociacion de datos :(");
    limpiarCajas();
}
```

Figura 49. Función de mandar información a la base de datos.

```
con = getConnection();
ps = con.prepareStatement(" UPDATE m_primas SET Date=?, Name_p=?, Cantidad=? WHERE Id_materia=?");
ps.setDate(1, Date.valueOf(Date.JTFC.getText()));
ps.setString(2, Name_p.JTFC.getText());
ps.setString(3, Cantidad.JTFC.getText());
ps.setString(4, (String) Id_materia.JCB.getSelectedItem());
int resultado = ps.executeUpdate();
```

Figura 50. Función para mandar nuevos datos y modificar los existentes en la base de datos.

```
con = getConnection();
ps = con.prepareStatement("DELETE FROM user WHERE Id_user=?");
ps.setInt(1, Integer.parseInt(Id_user.JTFC.getText()));
int res = ps.executeUpdate();
```

Figura 51. Función eliminar registro de la base de datos.

3.6 Producto Final

Después de las respectivas modificaciones a las interfaces y darle funcionalidad a todo el sistema, se realizó de nuevo una visita al cliente para que probara el producto final del software. En esta visita el cliente quedó satisfecho.

La funcionalidad del sistema se demostrará en el apartado 3.7 plan de pruebas para poner en evidencia el correcto funcionamiento del mismo.

Los apartados de grupo, categoría y unidad, no fueron requerimientos del usuario, pero se incluyeron con la finalidad de facilitar el cumplimiento de la regla número 1

de normalización de base de datos, la cual nos dice que una entidad sólo contiene datos propios.

3.7 Plan de pruebas

Se elaboró un plan de pruebas de caja negra referente a la funcionalidad del software. Se buscó poner en evidencia la calidad del software desarrollado, además de exponer los resultados finales obtenidos.

En las **tablas 11 y 12** se muestran las pruebas que se hicieron para verificar el funcionamiento de los “login” de los usuarios. Los resultados de dichas pruebas se observan en las **figuras 52 y 53** respectivamente.

| | | | |
|------------------------------------|--|----------------------------|----------------|
| Caso de prueba | Login gerente | | |
| Identificador caso de prueba | CP01_LogGer | | |
| Función probar | Login para gerentes. | | |
| Objetivo | Validar la funcionalidad del login del sistema para el restaurante “La Ventanita”. | | |
| Descripción | El usuario Gerente accede al sistema y selecciona el botón “Administrador”. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El usuario tipo gerente accede al menú para gerentes. | | |
| Criterios de falla | Lista de criterios de falla: 1. El sistema no lo deja acceder. 2. Se presenta mensaje informando que ocurrió un error. | | |
| Precondiciones | 1. Tener correo y contraseña de usuario tipo “Gerente” | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Correo: orla@cua.com, password: orla | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 29-11-2020 | | |
| | No paso | Usuario del sistema | Sistema |

| | | | |
|--------------------------|---|--|--|
| Flujo del caso de prueba | | | |
| | 1 | Entradas: Usuario y contraseña Descripción: El gerente entra al sistema y selecciona "Administrador". | El sistema presenta la pantalla de "Login gerente" |
| | 2 | Entradas: Usuario y contraseña. Descripción: Se llenan los formularios de usuario y contraseña. Se selecciona la opción "Iniciar". | El sistema le muestra el menú de gerentes |
| Post condiciones | El sistema ha dejado acceder al usuario | | |

Tabla 11. Caso de prueba 01 "Login usuario tipo gerente".

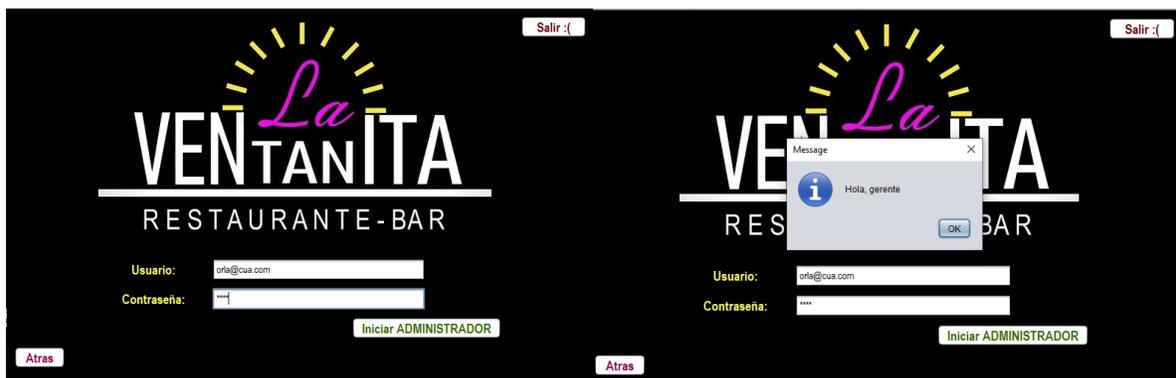


Figura 52. Resultado de prueba CP01_LogGer.

| | | |
|------------------------------|-----------|--|
| Caso de prueba | de | Login mesero |
| Identificador caso de prueba | | CP02_LogMes |
| Función probar | | Login para meseros. |
| Objetivo | | Validar la funcionalidad del login del sistema para el restaurante "La Ventanita". |
| Descripción | | El usuario Mesero accede al sistema y selecciona el botón "Iniciar". |
| Criterios de éxito | de | Lista de criterios de éxito: 1 El usuario tipo mesero accede al menú para meseros. |
| Criterios de falla | de | Lista de criterios de falla: 2 El sistema no lo deja acceder. 3 Se presenta mensaje informando que ocurrió un error. |
| Precondiciones | | 1. Tener correo y contraseña de usuario tipo "mesero". |

| | | | |
|------------------------------------|--|--|---|
| Perfil del usuario | Mesero. | | |
| Necesidades para el caso de prueba | Datos de prueba: Correo: portam12@gmail.com, password: Pass | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 29-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Entradas: Usuario y contraseña. | El sistema presenta la pantalla de mesero”. |
| | 2 | Entradas: Entradas: Usuario y contraseña. Descripción: Se llenan los formularios de usuario y contraseña. Se selecciona la opción “Iniciar”. | El sistema le muestra el menú de meseros. |
| Post condiciones | El sistema ha dejado acceder al usuario | | |

Tabla 12 Caso de prueba 02 “Login usuario tipo mesero”.



Figura 53. Resultado de prueba CP02_LogMes.

Las pruebas para las interfaces de “datos” se describen en **tablas 13, 14, 15 y 16**. Los resultados de las pruebas realizadas se visualizan en las **figuras 54, 55, 56 y 57** respectivamente.

| | | | |
|------------------------------------|---|---|--|
| Caso de prueba | Agregar datos de nuevo usuario | | |
| Identificador caso de prueba | CP03_AgrDatUsu | | |
| Función a probar | Agregar nuevos datos al sistema. | | |
| Objetivo | Dar de alta nuevo registro de datos. | | |
| Descripción | Dar de alta datos de un nuevo usuario | | |
| Criterios de éxito | Lista de criterios de éxito: 1. En el sistema se almacena la información. 2. Se muestra un mensaje de confirmación. | | |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra un mensaje en pantalla de error. | | |
| Precondiciones | 1. Conocer todos los datos del nuevo usuario. | | |
| Perfil del usuario | Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Nombre: Alberto Toledo Rojas, Numero: 2222389756, correo: albet@gmail.com, contraseña: rojastoledo | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 20-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Descripción: Entrar al apartado de agregar nuevos datos de usuario. | Muestra la interfaz de agregar datos de usuario. |
| | 2 | Entradas: Nombre, teléfono, correo y contraseña Descripción: Se añaden todos los datos en los campos correspondientes. | Se llenan los campos de información. |
| | 3 | Descripción: se presiona el botón de "Guardar". | Se muestra un mensaje de confirmación. |
| Post condiciones | En el sistema se ha almacenado la información del nuevo usuario. | | |

Tabla 13. Caso de prueba 03 "Agregar datos de nuevo usuario".



Figura 54. Resultado de prueba CP03_AgrDatUsu.

| Caso de prueba | Modificación de registro de información | | |
|------------------------------------|---|--|---|
| Identificador caso de prueba | CP04_ModDat | | |
| Función a probar | Modificación de información de usuario. | | |
| Objetivo | Validar la función de modificación de información de usuario. | | |
| Descripción | El usuario tipo gerente modifica alguno de los apartados de un registro de datos. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El sistema guarda las modificaciones. 2. Se muestra un mensaje en pantalla confirmando la acción realizada. | | |
| Criterios de falla | Lista de criterios de falla: 1. El registro no se guarda. 2. Se muestra un mensaje de error. | | |
| Precondiciones | Conocer el id de la información a modificar. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 3, contraseña: Orlando | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 20-11-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Descripción: Se ingresa al apartado de usuarios para después ingresar a "modificar información". | Se muestra la interfaz de modificar datos de usuario. |

| | | | |
|------------------|---|--|---|
| | 2 | Entradas: Id: 3 Descripción: Se selecciona el id a modificar y se realizan los cambios. | El sistema rellena los campos que es posible modificar. |
| | 3 | Descripción: Se presiona el botón de “modificar”. | Se muestra un mensaje de éxito de modificación. |
| Post condiciones | El sistema almacena los datos correctamente | | |

Tabla 14. Caso de prueba 04 “Modificar registro de datos”.



Figura 55. Resultado de prueba CP04_ModDat.

| Caso de prueba | de | Consultar registro de datos |
|------------------------------------|----|---|
| Identificador caso de prueba | | CP05_ConRegDat |
| Función a probar | | Tabla que muestra registro de datos. |
| Objetivo | | Validar el apartado de consulta de todos los registros de datos. |
| Descripción | | Se probará la función de consultar registro de datos. |
| Criterios de éxito | | Lista de criterios de éxito: 1. Se muestran los registros almacenados. |
| Criterios de falla | | Lista de criterios de falla 1. La tabla se queda con los valores predefinidos. |
| Precondiciones | | Ser usuario de con el perfil de gerente. |
| Perfil del usuario | | Gerente / Administrador. |
| Necesidades para el caso de prueba | | Datos de prueba: Ninguno. |
| Autor | | Miguel Angel Cuaya Tepeyahuitl. |

| | | | |
|--------------------------|---|--|--|
| Fecha de creación | 20-11-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Descripción: El usuario tipo gerente selecciona la opción de registro. | Se muestran todos los datos almacenados hasta ese momento. |
| Post condiciones | Se muestran todos los datos almacenados | | |

Tabla 15. Caso de prueba 05 “Consultar registros”.

| ID datos | Nombre | telefono | Correo | Contraseña |
|----------|--------------------------------|------------|--------------------|-------------|
| 1 | Miguel Angel Cuaya Tepeyahuitl | 2223016970 | portam12@gmail.com | Pass |
| 2 | Jacqueline Ramirez Dominguez | 2271034185 | jac@ho.com | domram |
| 3 | Orlando Cuautle Martinez | 2224218309 | oria@cua.com | Orlando |
| 4 | Alberto Toledo Rojas | 222389756 | alber@gmail.com | rojastoledo |

Figura 56. Resultado de prueba CP05_ConRegDat.

| | |
|------------------------------|---|
| Caso de prueba | Consulta de un registro de información |
| Identificador caso de prueba | CP06_ModDatUsu |
| Función a probar | Consultar información de un usuario. |
| Objetivo | Validar la función de consulta de información de un usuario. |
| Descripción | El usuario tipo gerente consulta un registro de datos. |
| Criterios de éxito | Lista de criterios de éxito: 1. El sistema muestra la información del registro seleccionado. |
| Criterios de falla | Lista de criterios de falla 1. No se muestra el registro de información. |
| Precondiciones | 1. Conocer el id de la información a consultar. |
| Perfil del usuario | Gerente / Administrador. |

| | | | |
|------------------------------------|---|---|---|
| Necesidades para el caso de prueba | Datos de prueba: Id: 1 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 20-11-2020 | | |
| | No paso | Flujo del caso de prueba | Sistema |
| Flujo del caso de prueba | 1 | Descripción: El usuario ingresa al apartado de usuarios para posteriormente seleccionar "información" | Se muestra el apartado de consulta de información |
| | 2 | Entradas: Id Descripción: Se selecciona la id de la cual se desea consultar la información. | El sistema muestra la información asociada la id seleccionada |
| Post condiciones | El sistema muestra los datos asociados. | | |

Tabla 16. Caso de prueba 06 Consulta de un registro de información.

Figura 57. Resultado de prueba CP06_ModDatUsu.

| | |
|------------------------------|--|
| Caso de prueba | Eliminar un registro de datos de usuario |
| Identificador caso de prueba | CP07_EliRegUsu |
| Función a probar | Función de eliminar un registro no asociado a un rol. |
| Objetivo | Eliminar a un usuario sin un rol. |
| Descripción | El usuario podrá eliminar un registro de datos sin un rol definido . |

| | | | |
|------------------------------------|---|--|---|
| Criterios de éxito | Lista de criterios de éxito: 1. El registro se elimina correctamente. | | |
| Criterios de falla | Lista de criterios de falla 1. Se muestra un mensaje de error. 2. El registro de datos ya está asociado a un rol. | | |
| Precondiciones | El registro no debe contar con un rol. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 4 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Descripción: Se debe ingresar al apartado usuarios y después seleccionar "consultar". | Muestra la interfaz de búsqueda. |
| | 2 | Entradas: ID: 4 Descripción: Se selecciona la id a consultar y posteriormente eliminar. | El sistema rellena los datos relacionados con la id seleccionada. |
| | 3 | Descripción: el usuario debe presionar el botón de "eliminar". | Se muestra un mensaje que confirma que el usuario ha sido borrado satisfactoriamente. |
| Post condiciones | El registro de datos se ha eliminado satisfactoriamente. | | |

Tabla 17. Caso de prueba 07 Eliminar usuario.



Figura 58. Resultado de prueba CP07_EliRegUsu.

Las pruebas para las interfaces de “asociar datos” se describen en **tablas 18, 19 y 20**. Los resultados de las pruebas realizadas se visualizan en las **figuras 59, 60 y 61** respectivamente.

| Caso de prueba | Asociar datos a un rol | | |
|------------------------------------|--|----------------------------|----------------|
| Identificador caso de prueba | CP08_AsoDat | | |
| Función a probar | Asociación de datos a un puesto de gerente o mesero. | | |
| Objetivo | Validar la función se asociación de datos. | | |
| Descripción | El usuario podrá asociar los datos de un registro a un rol dentro del sistema. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. La asociación es exitosa. 2. Se muestra un mensaje en pantalla que confirma la asociación. | | |
| Criterios de falla | Lista de criterios de falla 1. Se muestra en pantalla un mensaje de error. | | |
| Precondiciones | Ser usuario de tipo gerente. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: ID: 8, puesto: mesero | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |

| | | | |
|--------------------------|---------------------------|--|---|
| Flujo del caso de prueba | 1 | Entrar al apartado de usuarios y posteriormente seleccionar la opción de asociar. | Se muestra la interfaz de asociación de datos. |
| | 2 | Entradas: ID: 8, puesto: mesero Descripción: Seleccionar una id sin rol para asignarle uno y seleccionar su puesto. | Al seleccionar la Id correspondiente el sistema llenara datos referentes a ese usuario. |
| | 3 | Descripción: Se presiona el botón “guardar”. | El sistema muestra un mensaje en pantalla que confirma la asociación. |
| Post condiciones | La asociación fue exitosa | | |

Tabla 18 Caso de prueba 08 “Asociar usuario”.



Figura 59. Resultado de prueba CP08_AsoDat.

| | |
|------------------------------|---|
| Caso de prueba | Consultar registro de una asociación |
| Identificador caso de prueba | CP09_ConAEli |
| Función a probar | Consulta y eliminación de una asociación. |
| Objetivo | Consultar la asociación de un rol y eliminarla. |
| Descripción | El usuario podrá consultar y eliminar una asociación. |
| Criterios de éxito | Lista de criterios de éxito: 1. La consulta se hace satisfactoriamente. 2. La asociación es eliminada . |

| | | | |
|------------------------------------|---|---|---|
| Criterios de falla | Lista de criterios de falla: 1. La eliminación falla, por que el usuario ya desempeño tareas dentro del sistema. | | |
| Precondiciones | 1. Conocer el Id de una asociación que no cuente con tareas desempeñadas. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 17 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de usuarios para después seleccionar la opción de "consulta". | El sistema muestra la interfaz de "consulta de asociación". |
| | 2 | Entradas: Id: 17 Descripción: se selecciona una id que aún no tenga tareas desempeñadas en el sistema. | Se muestra información sobre su nombre, puesto e Id de datos. |
| | 3 | Descripción: Se selecciona la opción de "eliminar". | El sistema muestra un mensaje en pantalla confirma la acción realizada. |
| Post condiciones | El sistema ha eliminado la asociación de datos. | | |

Tabla 19. Caso de prueba 09 Búsqueda y eliminación de asociación.



Tabla 60. Resultado de prueba CP09_ConAEli.

| | | | |
|------------------------------------|--|---|--|
| Caso de prueba | Consulta de todos los usuarios con un rol específico | | |
| Identificador caso de prueba | CP010_CoAUsuRol | | |
| Función a probar | Función que muestra a todos los usuarios con un rol en el sistema. | | |
| Objetivo | Validar la interfaz que muestra a los usuarios dependiendo de su rol dentro del sistema. | | |
| Descripción | Se eligiera entre un puesto y se mostrara la información de los usuarios que tienen ese puesto. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestran los datos de los usuarios dependiendo de su rol dentro del sistema. | | |
| Criterios de falla | Lista de criterios de falla 1. No se muestra nada. | | |
| Precondiciones | Tener usuarios suficientes que tengan asignado un rol. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: ninguno. | | |
| Autor | Miguel Angel Cuaya Tepeyahuítl | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresará al menú de usuarios para después seleccionar la opción de "Empleados". | Se muestra una interfaz en la se escogerá el perfil de usuario que se desea consultar. |
| | 2 | Detalles: El usuario selecciona el perfil a consultar. | Se muestra información de los usuarios que están asignados al rol seleccionado . |
| Post condiciones | Se muestra la información de los usuarios almacenados dependiendo del rol que desempeñan. | | |

Tabla 20. Caso de prueba 10. Consulta dependiendo del Rol.



Figura 61 Resultado de prueba CP010_CoAUsuRol

La prueba para la interfaz de “ganancias” se describe en **tabla 21**. El resultado de la prueba realizada se visualiza en la **figura 62**.

| | | | |
|------------------------------------|--|--|--|
| Caso de prueba | Ganancias | | |
| Identificador caso de prueba | CP11_Gan | | |
| Función a probar | Calcular ganancias. | | |
| Objetivo | Validar la función de calcular ganancias. | | |
| Descripción | Se seleccionará una fecha en la cual existan registros de órdenes y el sistema calculará las ganancias de ese día. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El sistema calcula las ganancias de un día específico. | | |
| Criterios de falla | Lista de criterios de falla 1. El sistema no calcula nada. | | |
| Precondiciones | Deben existir registros de ventas. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: 2020-11-17 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de usuarios para después seleccionar “ganancias”. | Se muestra una interfaz con un botón donde están las fechas que tienen registros de órdenes. |

| | | | |
|------------------|--|---|---|
| | 2 | Entradas: fecha 2020-11-17 Descripción: Se selecciona la fecha . | Se calcula el total de todas las ventas del día seleccionado. |
| Post condiciones | Se conoce el total de ganancias del día seleccionado | | |

Tabla 21. Caso de prueba 11. Ganancias.



Figura 62. Resultado de prueba CP11_Gan.

Las pruebas para las interfaces de “mensajes para meseros” se describen en **tablas 22 y 23**. Los resultados de las pruebas realizadas se visualizan en las **figuras 63 y 64** respectivamente.

| Caso de prueba | Mensaje para meseros |
|------------------------------|--|
| Identificador caso de prueba | CP12_MenParMes |
| Función a probar | Función de almacenar mensaje para meseros. |
| Objetivo | Demostrar el correcto funcionamiento de almacenamiento de mensajes para meseros. |
| Descripción | El usuario tipo gerente genera un mensaje que posteriormente los meseros podrán consultar. |
| Criterios de éxito | Lista de criterios de éxito: <ol style="list-style-type: none"> 1. El mensaje se almacena correctamente. 2. Se muestra una alerta en pantalla que confirma el almacenamiento correcto del mensaje. 3. El mensaje es posible ver en el apartado de mensajes. |
| Criterios de falla | Lista de criterios de falla: <ol style="list-style-type: none"> 1. El mensaje no es almacenado 2. No se puede visualizar en el apartado de “mensajes por fecha”. |

| | | | |
|------------------------------------|---|--|--|
| Precondiciones | Ser usuario con el rol de gerente. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 7, Fecha 2020-11-30, mensaje: "Hola, esta es una prueba" | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Se ingresa al apartado mensajes y se selecciona "nuevo mensaje para meseros". | Se muestra la interfaz "nuevo mensaje para meseros". |
| | 2 | Entradas: Id: 7, fecha 2020-11-30, mensaje: "Hola, esta es una prueba" Descripción: Se elige la Id del gerente que realiza el mensaje, posteriormente se llenan los apartados correspondientes. Se presiona el botón de "generar nota". | Se muestra un mensaje en pantalla confirmando que el mensaje fue almacenado correctamente. |
| Post condiciones | El mensaje debe poder verse en el apartado de mensajes por fecha. | | |

Tabla 22. Caso de prueba 12 Generar mensaje.



Figura 63. Resultado de prueba CP12_MenParMes.

| | | | |
|------------------------------------|---|---|---|
| Caso de prueba | Consultar mensajes de un día específico | | |
| Identificador caso de prueba | CP13_ConMenDía | | |
| Función a probar | Validar la función de consulta de mensajes por día. | | |
| Objetivo | Mostrar los mensajes almacenados en la base de datos. | | |
| Descripción | El usuario será capaz de seleccionar un día en el que se hayan registrado mensajes y podrá ver los mensajes de ese día. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestran los mensajes del día seleccionado. | | |
| Criterios de falla | Lista de criterios de falla: 1. No se muestra nada en pantalla por un error de codificación. | | |
| Precondiciones | Conocer el día en el que se realizaron registros de mensajes. | | |
| Perfil del usuario | Gerente / Mesero. | | |
| Necesidades para el caso de prueba | Datos de prueba: Fecha: 2020-12-01 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Ingresar al apartado "registro de mensajes". | Mostrará la interfaz en donde se seleccionará la fecha a consultar . |
| | 2 | Entradas: Fecha: 2020-11-30 Descripción: Se selecciona la fecha. | Se muestra en pantalla todos los mensajes que los usuarios "gerente" generaron. |
| Post condiciones | Se muestra en pantalla el mensaje, nombre e identificador de usuario de la persona que realizo el mensaje. | | |

Tabla 23. Consultar mensajes por fecha.



Figura 64. Resultado de prueba CP13_ConMenDía.

Las pruebas para las interfaces de “órdenes” se describen en **tablas 24, 25 y 26**. Los resultados de las pruebas realizadas se visualizan en las **figuras 65, 66 y 67** respectivamente.

| | | | |
|------------------------------------|--|----------------------------|----------------|
| Caso de prueba | Calcular total de una orden | | |
| Identificador caso de prueba | CP14_CalTotOrd | | |
| Función a probar | Función de calcular el total de una cuenta registrada en el sistema. | | |
| Objetivo | Conocer el total a pagar de una cuenta en específico. | | |
| Descripción | El usuario podrá guardar el total de una cuenta existente. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestra en la interfaz el total de la cuenta seleccionada. | | |
| Criterios de falla | Lista de criterios de falla 1. No se hace el cálculo correspondiente. | | |
| Precondiciones | La orden seleccionada debe tener artículos asociados a él. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id:2 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| | | | |

| | | | |
|--------------------------|---|--|---|
| Flujo del caso de prueba | 1 | El usuario tipo gerente ingresa al apartado "ordenes". | Se muestran las opciones de cobrar, consultar y calcular total. |
| | 2 | Entradas: ID: 2 Descripción: El usuario selecciona la opción de cobrar cuenta, selecciona entre las opciones el ID 2. | Muestra el nombre de quien atendió esa orden, la fecha de realización y la nota para cocina registrada. |
| | 3 | Descripción: Se presiona el botón de calcular y posteriormente el de modificar. | El total fue calculado y almacenado correctamente. |
| Post condiciones | Se debe poder tomar en cuenta este registro en el siguiente cálculo de ganancias. | | |

Tabla 24. Caso de prueba 14 Calcular total de una cuenta.



Figura 65. Resultado de prueba CP14_CalTotOrd

| | |
|------------------------------|---|
| Caso de prueba | Ver pedidos de una orden |
| Identificador caso de prueba | CP15_PedOrd |
| Función a probar | Consulta de platillos asociados a una orden. |
| Objetivo | Conocer los datos asociados a una orden. |
| Descripción | Se desea conocer que platillos pidió un comensal en su orden. |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestran todos los platillos que pidió un comensal en su orden. |

| | | | |
|------------------------------------|---|---|--|
| Criterios de falla | Lista de criterios de falla: 1. No se muestra la información consultada. | | |
| Precondiciones | 1. Que la orden cuente con platillos en su registro. 2. Conocer el id de la orden. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: ID: 2 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | Se accede al menú "órdenes" desde el menú para gerentes. Posteriormente se selecciona la opción de "registros". | Se muestran las opciones de cobrar, consultar lista de platillos de una orden y cobrar cuenta. |
| | 2 | Entradas: ID 2 Descripción: Se selecciona la id en el botón de búsqueda. | Se muestran todos los platillos relacionados al id seleccionado. |
| Post condiciones | Se conoce los platillos que fueron registrados en el sistema. | | |

Tabla 25. Caso de prueba 15 Consultar registro de orden.



Figura 66. Resultado de prueba CP15_PedOrd.

| | | | |
|------------------------------------|--|--|--|
| Caso de prueba | Cobrar cuenta de una orden | | |
| Identificador caso de prueba | CP16_CobCue | | |
| Función a probar | Validar el correcto funcionamiento de la función de cobrar cuenta. | | |
| Objetivo | Comprobar la funcionalidad del apartado de cobrar cuenta. | | |
| Descripción | Se seleccionara un registro de orden del cual previamente se calculó el total. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El cobro se hace satisfactoriamente 2. El registro queda almacenado con el descuento (si lo amerita) | | |
| Criterios de falla | Lista de criterios de falla: 1. No se puede almacenar el total de la cuenta. | | |
| Precondiciones | De la cuenta que se desea cobrar, se debió haber calculado el total. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: ID:2, descuento:5, recibes:200. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 30-11-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario accede al apartado de cobrar cuenta desde el menú de órdenes. | Muestra la interfaz que corresponde al cobro de cuenta. |
| | 2 | Entradas: ID:2 y descuento:5 Descripción: Se selecciona la Id correspondiente y se llena el espacio de "descuento". | Al seleccionar la id de orden que se desea cobrar se llenarán los espacios de nombre de mesero que atendió, nota para cocina y la fecha. |
| | 3 | Entradas: recibes:200 Descripción: se rellena el espacio de descuento y se calcula el total, posteriormente se ingresa la cantidad de dinero recibido y se calcula el dinero que se devolverá . | Se calcula el total con el descuento (si lo amerita) y posteriormente se muestra el total de cambio a devolver |

| | | | |
|------------------|--|--|---|
| | | | Se muestra el mensaje de cobro exitoso. |
| Post condiciones | El registro de la orden queda modificado con el total de la cuenta | | |

Tabla 26. Caso de prueba 16 Cobrar cuenta.

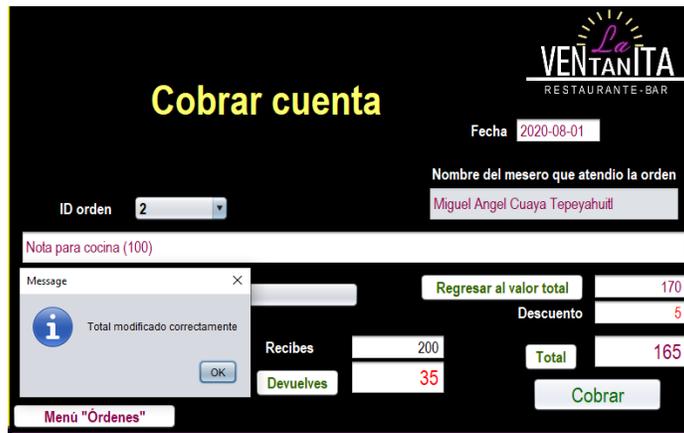


Figura 67. Resultado de prueba CP15_CobCue

Las pruebas para las interfaces de “platos” se describen en **tablas 27, 28, 29 y 30**. Los resultados de las pruebas realizadas se visualizan en las **figuras 68, 69, 70, 71, 72 y 73** respectivamente. Nota: Las figuras 70, 71 y 72 pertenecen a un caso de prueba.

| Caso de prueba | de | Agregar producto a Menú |
|------------------------------|----|--|
| Identificador caso de prueba | | CP17_NueProMen |
| Función a probar | | Función de agregar nuevo producto al menú . |
| Objetivo | | Agregar un platillo a la carta. |
| Descripción | | Se agregará un nuevo producto disponible a la carta. |
| Criterios de éxito | de | Lista de criterios de éxito: 1. El nombre y precio del producto agregado aparece el apartado de consulta de platos. |
| Criterios de falla | de | Lista de criterios de falla 1. El registro no se lleva a cabo mostrando un mensaje en pantalla de error |
| Precondiciones | | Se deben conocer los datos del platillo a agregar |

| | | | |
|------------------------------------|--|---|---|
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Nombre: Rebanadas de pastel de chocolate, precio: 15, Id grupo: 8 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 30-11-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de nuevo producto desde el menú "carta". | Muestra la interfaz de "nuevo platillo de la carta". |
| | 2 | Entradas: Nombre: Rebanadas de pastel de chocolate, precio: 15, Id grupo: 8 Descripción: El usuario llena los apartados del formulario de nombre, precio y selecciona el grupo al que pertenece el platillo. | Al seleccionar el grupo se llena el campo de nombre de grupo con el fin de que el usuario conozca el grupo al que agregara el producto. |
| | 3 | Descripción: Se presiona el botón de "guardar". | El sistema muestra un mensaje de que el registro se realizó correctamente. |
| Post condiciones | El platillo agregado se visualiza en su grupo correspondiente. | | |

Tabla 27. Caso de prueba 17 Agregar producto a la carta.



Figura 68. Resultado de prueba CP17_NueProMen.

| Caso de prueba | Modificación de información d un platillo de la carta | | |
|------------------------------------|--|--|---|
| Identificador caso de prueba | CP18_ModDatPlat | | |
| Función a probar | Modificación de información de platillo. | | |
| Objetivo | Cambiar la información de un platillo de la carta. | | |
| Descripción | Se busca probar la función de modificación de un producto de la carta | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El nombre o grupo son modificados correctamente. | | |
| Criterios de falla | Lista de criterios de falla: 1. La modificación no se lleva a cabo mostrando un mensaje de error en pantalla. | | |
| Precondiciones | Conocer el dato que se desea modificar (nombre). | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 6, grupo: 15 (TSFDS: sólo fin de semana). | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de modificar desde el menú "carta". | Se muestra la interfaz de "modificar producto de la carta". |
| | 2 | Entradas: Id: 6 y grupo: 15 | En el botón de selección de |

| | | | |
|------------------|--|--|---|
| | | Descripción: Se selecciona la id del producto que se desea modificar, y se cambia el grupo al que pertenece. | identificador de platillo deben aparecer todos los platillos agregados hasta ese momento. |
| | 3 | Descripción: Se presiona el botón de “guardar”. | Se muestra un mensaje en pantalla confirmando la modificación del producto. |
| Post condiciones | Las modificaciones se ven reflejadas si volvemos a seleccionar el producto | | |

Tabla 28. Caso de prueba 18 Modificar datos de un platillo del menú.



Figura 69. Resultado de prueba CP18_ModDatPlat.

| Caso de prueba | de | Consulta de platillos por grupo |
|------------------------------|----|--|
| Identificador caso de prueba | | CP19_ConPlaGru |
| Función a probar | | Consulta de platillos por grupos de menú. |
| Objetivo | | Validar la función de consulta para los platillos. |
| Descripción | | Consultar los platillos la categoría a la que pertenecen. |
| Criterios de éxito | de | Lista de criterios de éxito: 1. El platillo es mostrado en la categoría que le corresponde. |
| Criterios de falla | de | Lista de criterios de falla: 1. El platillo está en un categoría que no es la suya. |
| Precondiciones | | Deben de existir registros de platillos en el sistema. |

| | | | |
|------------------------------------|--|---|--|
| Perfil del usuario | Gerente o Mesero. | | |
| Necesidades para el caso de prueba | Datos de prueba: ninguno. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | | | |
| | 1 | Detalles: el usuario accede al apartado de registro de datos. | Muestra la interfaz con un botón de selección . |
| | 2 | Detalles: El usuario navega entre los diferentes grupos de platillos. | El sistema muestra los platillos correspondientes al grupo seleccionado. |
| Post condiciones | El sistema debe mostrar los datos de los platillos de la categoría seleccionada. | | |

Tabla 29. Caso de prueba 19 Consulta de grupos de platillos.



Figura 70. Resultado 1 de prueba CP19_ConPlaGru.



Figura 71. Resultado 2 de prueba CP19_ConPlaGru.



Figura 72. Resultado 3 de prueba CP19_ConPlaGru.

| | | | |
|------------------------------------|--|---|--|
| Caso de prueba | Consultar platillo de menú | | |
| Identificador caso de prueba | CP20_ConPlaMen | | |
| Función a probar | Consulta de un platillo del menú. | | |
| Objetivo | Buscar un platillo de menú. | | |
| Descripción | Se busca validar la consulta de un producto del menú. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. La id seleccionada muestra los datos del platillo a eliminar. | | |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra un mensaje en pantalla de error. | | |
| Precondiciones | Conocer el id del platillo a consultar. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 5 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 30-11-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de consulta de información del menú "carta". | Se muestra la interfaz en donde se buscará el platillo a consultar y/o eliminar. |
| | 2 | Entradas: Id: 5 Descripción: El usuario selecciona la ID del platillo. | Se muestra la información del platillo seleccionado. |
| Post condiciones | El información del platillo es mostrada. | | |

Tabla 30. Caso de prueba 20 Consultar información de un platillo.



Figura 73. Resultado de prueba CP20_ConPlaMen.

Las pruebas para las interfaces de “grupos” se describen en **tablas 31, 32, 33 y 34**. Los resultados de las pruebas realizadas se visualizan en las **figuras 74, 75, 76 y 77** respectivamente.

| Caso de prueba | Agregar nuevo grupo de menú | | |
|------------------------------------|---|----------------------------|----------------|
| Identificador caso de prueba | CP21_AgrNueGruMen | | |
| Función a probar | Interfaz de agregar nuevo grupo. | | |
| Objetivo | Agregar un nuevo grupo de menú | | |
| Descripción | Se busca validar la función de agregar nuevo grupo para menú. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El grupo se agregó al sistema. 2. Se muestra un mensaje en pantalla que confirma que el grupo fue agregado correctamente. | | |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra un mensaje de error en pantalla. 2. El grupo ya existe. | | |
| Precondiciones | Contar con la información del grupo que se desea agregar. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Nombre: FASTeat | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |

| | | | |
|--------------------------|---|--|---|
| Flujo del caso de prueba | 1 | El usuario selecciona la opción de “nuevo grupo” desde el menú “carta”. | Muestra el formulario para asignarle un nombre al nuevo grupo. |
| | 2 | Entradas: Nombre: FASTeat Descripción: Se llena el apartado de nombre y se procede a presionar el botón de guardar. | Se muestra un mensaje en pantalla que confirma el registro del nuevo grupo. |
| Post condiciones | El nuevo grupo se muestra en las opciones de selección de grupo para platillos. | | |

Tabla 31. Caso de prueba 21 Agregar nuevo grupo.



Figura 74. Resultado de prueba CP21_AgrNueGruMen.

| Caso de prueba | de | Listado de grupos |
|------------------------------|----|---|
| Identificador caso de prueba | | CP22_RegGru |
| Función a probar | | Interfaz donde se muestran todos los grupos . |
| Objetivo | | Verificar el correcto funcionamiento de la interfaz. |
| Descripción | | Se ingresará al apartado “listado de grupos” para ver todos los grupos almacenados en el sistema. |
| Criterios de éxito | de | Lista de criterios de éxito: 1. Se muestran todos los grupos almacenados en la base de datos. |
| Criterios de falla | de | Lista de criterios de falla 1. No se muestra nada en la interfaz. |
| Precondiciones | | Se debe tener registrado al menos un grupo en el sistema. |

| | | | |
|------------------------------------|---|---|--|
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Ninguno. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario accede al apartado de "listado de grupos". | Muestra todos los grupos que se han dado de alta hasta ese momento |
| Post condiciones | Se muestran el id y descripción de los grupos | | |

Tabla 32. Caso de prueba 22 Listado de grupos.



Figura 75. Resultado de prueba CP22_RegGru.

| | |
|------------------------------|---|
| Caso de prueba | Modificar nombre de un grupo |
| Identificador caso de prueba | CP23_ModNomGru |
| Función a probar | Interfaz de cambio de nombre de grupo. |
| Objetivo | Cambiar el nombre de un grupo específico |
| Descripción | Se busca cambiar el nombre de un grupo existente |
| Criterios de éxito | Lista de criterios de éxito: 1. El nombre del grupo se modifica correctamente. 2. Se muestra un mensaje en pantalla que confirma la modificación. |

| | | | |
|------------------------------------|--|--|--|
| Criterios de falla | Lista de criterios de falla: 1. El nuevo nombre ya pertenece a una entidad. 2. Se muestra una alerta de error en pantalla. | | |
| Precondiciones | Que el nuevo nombre no este ocupado por un grupo | | |
| Perfil del usuario | Gerente / Administrador | | |
| Necesidades para el caso de prueba | Datos de prueba: Id:7, nombre: Parrilladas | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de modificar dentro del menú carta el apartado de grupo . | Muestra la interfaz de modificación. |
| | 2 | Entradas: Id: 7, nombre: Parrilladas Descripción: Seleccionamos la id del grupo a modificar; cambiamos el nombre. | El programa llena el campo con el nombre que le corresponde al grupo seleccionado. |
| | 3 | Descripción: Presionamos el botón de “modificar”. | Se muestra un mensaje de confirmación de la modificación. |
| Post condiciones | Al seleccionar de nuevo la Id aparece con los datos nuevos | | |

Tabla 33. Caso de prueba 23 Modificación de grupo.

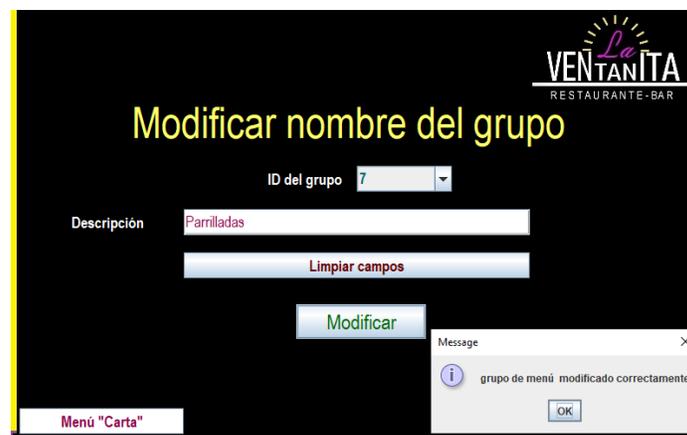


Figura 76. Resultado de prueba CP23_ModNomGru.

| | | | |
|------------------------------------|--|--|--|
| Caso de prueba | Consulta y eliminación de un grupo almacenado en el sistema | | |
| Identificador caso de prueba | CP24_ConEliGru | | |
| Función a probar | Buscar y eliminar un grupo del sistema. | | |
| Objetivo | Eliminar grupo que no tenga datos asociados a el. | | |
| Descripción | Se eliminará un grupo que no cuente con un platillo asociado a el. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestra un mensaje en pantalla que confirma la eliminación del producto. 2. El grupo ya no aparece en el botón de búsqueda. | | |
| Criterios de falla | Lista de criterios de falla: 1. El grupo ya está asociado a un platillo. 2. Se muestra un mensaje de error en pantalla. | | |
| Precondiciones | El grupo que se desea eliminar no debe estar asociado a un platillo. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 7 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de consulta en el menú de carta en la sección de grupo. | Muestra la interfaz de consulta y eliminación de un grupo. |
| | 2 | Entradas: ID:7 Descripción: Se selecciona la Id del grupo que se desea borrar. | Se rellena el campo de descripción. |
| | 3 | Descripción: Se oprime el botón de “eliminar”. | Aparece un mensaje en pantalla que confirma la eliminación del grupo . |
| Post condiciones | La id borrada ya no aparece entre las opciones seleccionables. | | |

Tabla 34. Caso de prueba 24 Consulta y eliminación de grupo.



Figura 77. Resultado de prueba CP24_ConEliGru.

Las pruebas para las interfaces de “unidades” se describen en **tablas 26, 27, 28 y 29**. Los resultados de las pruebas realizadas se visualizan en las **figuras 78, 79, 80 y 81** respectivamente.

| Caso de prueba | Agregar nueva unidad de medición para inventario |
|------------------------------------|---|
| Identificador caso de prueba | CP25_AgrNueUnilnv |
| Función a probar | Interfaz de agregar nueva unidad. |
| Objetivo | Agregar una nueva unidad de medición para inventario. |
| Descripción | Se busca validar la función de agregar nueva unidad de medida de inventario. |
| Criterios de éxito | Lista de criterios de éxito: 1. La unidad se agregó al sistema. 2. Se muestra un mensaje en pantalla de que la unidad fue agregada correctamente. |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra un mensaje de error en pantalla. 2. La unidad de medida ya existe. |
| Precondiciones | Contar con la información de la unidad de media que se desea agregar. |
| Perfil del usuario | Gerente / Administrador. |
| Necesidades para el caso de prueba | Datos de prueba: Nombre: Onzas |
| Autor | Miguel Angel Cuaya Tepeyahuitl. |

| | | | |
|--------------------------|---|---|--|
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario selecciona la opción de “nueva unidad” desde el menú “inventario”. | Muestra el formulario para asignarle un nombre a la nueva unidad. |
| | 2 | Entradas: Nombre: Onza Descripción: Se llena el apartado de nombre y se procede a presionar el botón de guardar. | Se muestra un mensaje en pantalla que confirma el registro de la nueva unidad. |
| Post condiciones | La nueva unidad se muestra en las opciones de selección de grupo para inventario. | | |

Tabla 35. Caso de prueba 25 Agregar nueva unidad.



Figura 78. Resultado de prueba CP25_AgrNueUniInv.

| | |
|------------------------------|---|
| Caso de prueba | Listado de unidades |
| Identificador caso de prueba | CP26_RegUni |
| Función a probar | Interfaz donde se muestran todas las unidades de medición. |
| Objetivo | Verificar el correcto funcionamiento de la interfaz. |
| Descripción | Se ingresará al apartado “listado de unidades” para ver todas las unidades almacenadas en el sistema. |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestran todas unidades almacenados en la base de datos. |

| | | | |
|------------------------------------|--|---|---|
| Criterios de falla | Lista de criterios de falla: 1. No se muestra nada en la interfaz. | | |
| Precondiciones | Se debe tener registrada al menos una unidad en el sistema. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Ninguno. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | | | |
| | 1 | El usuario accede al apartado de "registro" en la sección unidades. | Muestra todas las unidades que se han almacenado hasta ese momento. |
| Post condiciones | Se muestran el id y la descripción de las unidades de medida del inventario. | | |

Tabla 36. Caso de prueba 26 Registro de unidades.

| ID | Descripcion |
|----|-------------|
| 1 | Botella |
| 2 | Bolsa |
| 3 | Barra |
| 4 | Kilo |
| 5 | Bote |
| 6 | Unidad |
| 7 | Litro |
| 8 | Manojos |
| 11 | Manojos |
| 15 | Onzas |

Figura 79. Resultado de prueba CP26_RegUni.

| | |
|------------------------------|---|
| Caso de prueba | Modificar nombre de una unidad |
| Identificador caso de prueba | CP27_ModNomUni |
| Función a probar | Interfaz de cambio de nombre de unidad de medida. |
| Objetivo | Cambiar el nombre de una unidad específica. |

| | | | |
|------------------------------------|--|--|---|
| Descripción | Se busca cambiar el nombre de una unidad existente. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El nombre de la unidad se modifica correctamente. 2. Se muestra un mensaje en pantalla que confirma la modificación. | | |
| Criterios de falla | Lista de criterios de falla: 1. El nuevo nombre ya pertenece a una unidad registrada. 2. Se muestra una alerta de error en pantalla. | | |
| Precondiciones | Que el nuevo nombre no este ocupado por una unidad de medición registrada. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id:16, nombre: Miligramos | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de “modificar” dentro del “menú inventario” en el apartado de “unidades”. | Muestra la interfaz de modificación. |
| | 2 | Entradas: Id: 16, nombre: Miligramos. Descripción: Seleccionamos la id de la unidad a modificar; cambiamos el nombre. | El programa llena el campo con el nombre que le corresponde a la unidad seleccionada. |
| | 3 | Descripción: Presionamos el botón de “modificar”. | Se muestra un mensaje de confirmación de la modificación. |
| Post condiciones | Al seleccionar de nuevo la Id ya aparece con los nuevos datos | | |

Tabla 37. Caso de prueba 27 Modificación de una unidad.



Figura 80. Resultado de prueba CP27_ModNomUni.

| Caso de prueba | Consulta y eliminación de una unidad de medida almacenada en el sistema | | |
|------------------------------------|---|----------------------------|----------------|
| Identificador caso de prueba | CP28_ConEliUni | | |
| Función a probar | Buscar y eliminar una unidad de medida almacenada en el sistema. | | |
| Objetivo | Eliminar unidad de medida que no tenga datos asociados a ella. | | |
| Descripción | Se eliminará la unidad que no cuente con un registro asociado a ella. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestra un mensaje en pantalla que confirma la eliminación de la unidad. 2. La unidad ya no aparece en el botón de búsqueda. | | |
| Criterios de falla | Lista de criterios de falla 1. La unidad está asociada a un material del inventario. 2. Se muestra un mensaje de error en pantalla. | | |
| Precondiciones | La unidad que se desea eliminar no debe estar asociada a un registro de inventario. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 16 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |

| | | | |
|--------------------------|---|--|--|
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de consulta en el menú de inventario en la sección de unidades. | Muestra la interfaz de consulta y eliminación de unidades. |
| | 2 | Entradas: ID:16 Descripción: El usuario selecciona la Id de la unidad que se desea borrar. | Se rellena el campo de descripción. |
| | 3 | Descripción: Se oprime el botón de “eliminar”. | Aparece un mensaje en pantalla que confirma la eliminación de la unidad. |
| Post condiciones | El id que se eliminó ya no aparece entre las opciones seleccionables. | | |

Tabla 38. Caso de prueba 28 Consulta y eliminación de unidad.

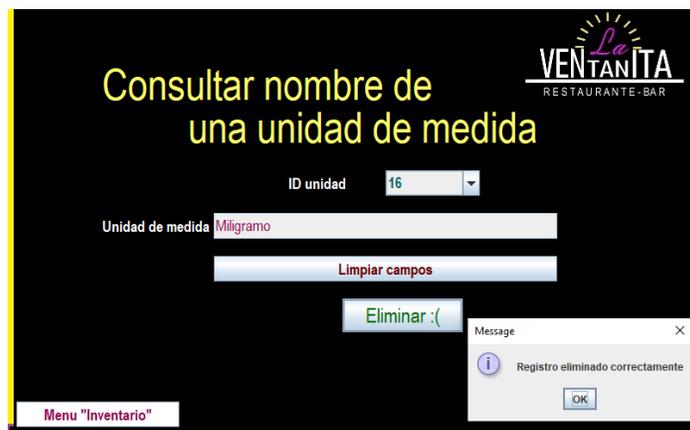


Figura 81. Resultado de prueba CP28_ConEliUni.

Las pruebas para las interfaces de “categoría” se describen en **tablas 39, 40, 41 y 42**. Los resultados de las pruebas realizadas se visualizan en las **figuras 82, 83, 84 y 85** respectivamente.

| | | |
|------------------------------|-----------|--|
| Caso de prueba | de | Agregar nueva categoría para inventario |
| Identificador caso de prueba | | CP29_AgrNueCatInv |
| Función a probar | | Interfaz de agregar nueva categoría. |
| Objetivo | | Agregar una nueva categoría para inventario. |

| | | | |
|------------------------------------|--|--|---|
| Descripción | Se busca validar la función de agregar nueva categoría de inventario. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. La categoría se agregó al sistema satisfactoriamente. 2. Se muestra un mensaje en pantalla confirmando que la categoría fue agregada correctamente al sistema. | | |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra un mensaje de error en pantalla. 2. La categoría ya existe en el sistema. | | |
| Precondiciones | Contar con la información de la categoría que se desea agregar. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Nombre: Utensilios. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 30-11-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario selecciona la opción de “nueva categoría” desde el menú “inventario”. | Muestra el formulario para asignarle un nombre a la nueva categoría. |
| | 2 | Entradas: Nombre: Utensilios Descripción: Se llena el apartado de descripción y se procede a presionar el botón de guardar. | Se muestra un mensaje en pantalla que confirma el registro de la nueva categoría de inventario. |
| Post condiciones | La nueva categoría se muestra en las opciones de selección de grupo para inventario | | |

Tabla 39. Caso de prueba 29 Agregar nueva categoría.

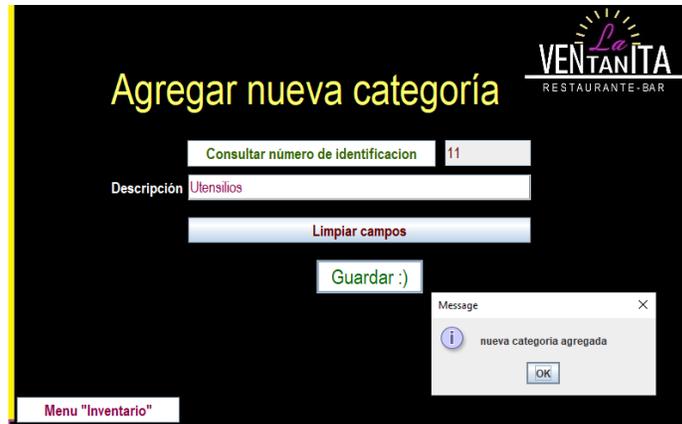


Figura 82. Resultado de prueba CP29_AgrNueCatInv.

| Caso de prueba | Listado de categoría | | |
|------------------------------------|---|---|---|
| Identificador caso de prueba | CP30_RegCat | | |
| Función a probar | Interfaz donde se muestran todas las categorías registradas. | | |
| Objetivo | Verificar el correcto funcionamiento de la interfaz. | | |
| Descripción | Se ingresará al apartado "listado de categoría" para ver todas las categorías almacenadas en el sistema | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestran todas categoría almacenados en la base de datos hasta ese momento. | | |
| Criterios de falla | Lista de criterios de falla: 1. No se muestra nada en la interfaz. | | |
| Precondiciones | Se debe tener registrada al menos una categoría en el sistema. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Ninguno. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | | | |
| | 1 | El usuario accede al apartado de "listado | Muestra todas las categorías que se han |

| | | | |
|------------------|--|--|---------------------------------|
| | | de categorías” en la sección categorías. | dado de alta hasta ese momento. |
| Post condiciones | Se muestra el id y descripción de las categorías del inventario. | | |

Tabla 40. Caso de prueba 30 Registro de categorías.



Figura 83. Resultado de prueba CP30_RegCat.

| | |
|------------------------------|--|
| Caso de prueba | Modificar nombre de una categoría |
| Identificador caso de prueba | CP31_ModNomCat |
| Función a probar | Interfaz de cambio de nombre de categoría. |
| Objetivo | Cambiar el nombre de una categoría específica. |
| Descripción | Se busca cambiar el nombre de una categoría registrada en el sistema. |
| Criterios de éxito | Lista de criterios de éxito: 1. El nombre de la categoría se ha modificado correctamente. 2. Se muestra un mensaje en pantalla que confirma la modificación. |
| Criterios de falla | Lista de criterios de falla: 1. El nuevo nombre ya pertenece a una categoría registrada. 2. Se muestra una alerta de error en pantalla. |
| Precondiciones | Que el nuevo nombre no este ocupado por una categoría registrada. |
| Perfil del usuario | Gerente / Administrador. |

| | | | |
|------------------------------------|--|---|---|
| Necesidades para el caso de prueba | Datos de prueba: Id: 9, nombre: Cucharas. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de “modificar” dentro del “menú inventario” en el apartado de “categoría”. | Muestra la interfaz de modificación. |
| | 2 | Entradas: Id:9, nombre: Cucharas. Descripción: Seleccionamos la id de la categoría a modificar; cambiamos el nombre. | El programa llena el campo con la descripción que le corresponde a la categoría seleccionada. |
| | 3 | Descripción: Presionamos el botón de “modificar”. | Se muestra un mensaje de confirmación de la modificación. |
| Post condiciones | Al seleccionar de nuevo la Id ya aparece con los nuevos datos. | | |

Tabla 41. Caso de prueba 31 Modificación de una categoría.

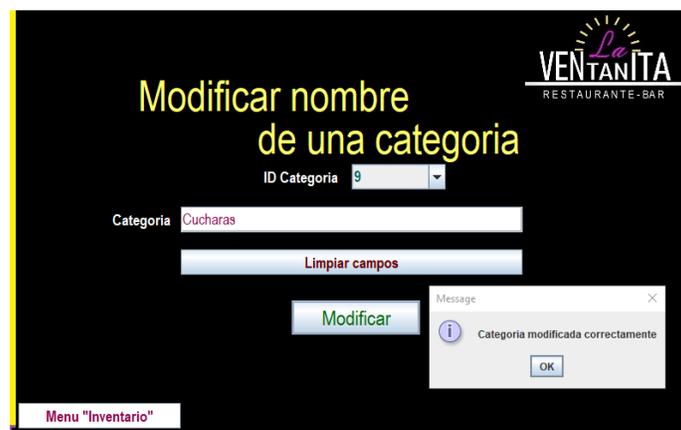


Figura 84. Resultado de prueba CP31_ModNomCat.

| | | | |
|------------------------------------|---|---|---|
| Caso de prueba | Consulta y eliminación de una categoría de medida almacenada en el sistema | | |
| Identificador caso de prueba | CP32_ConEliCat | | |
| Función a probar | Buscar y eliminar una categoría almacenada en el sistema. | | |
| Objetivo | Eliminar categoría que no tenga datos asociados a ella. | | |
| Descripción | Se eliminará una categoría siempre y cuando no cuente con un registro asociado a ella. | | |
| Criterios de éxito | Lista de criterios de éxito: 1 Se muestra un mensaje en pantalla que confirma la eliminación de la categoría. 2 La categoría ya no aparece en el botón de búsqueda. | | |
| Criterios de falla | Lista de criterios de falla: 1. La categoría está asociada a un material del inventario. 2. Se muestra un mensaje de error en pantalla. | | |
| Precondiciones | La categoría que se desea eliminar no debe estar asociada a un registro de inventario. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 9 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de consulta en el menú de "inventario" en la sección de "categoría". | Muestra la interfaz de consulta y eliminación de categorías. |
| | 2 | Entradas: ID:9 Descripción: El usuario selecciona la Id de la categoría que desea borrar. | Se rellena el campo de descripción. |
| | 3 | Descripción: Se oprime el botón de "eliminar". | Aparece un mensaje en pantalla que confirma la eliminación de la categoría. |
| Post condiciones | El id que se eliminó ya no aparece entre las opciones seleccionables. | | |

Tabla 42. Caso de prueba 32 Consulta y eliminación de categoría.

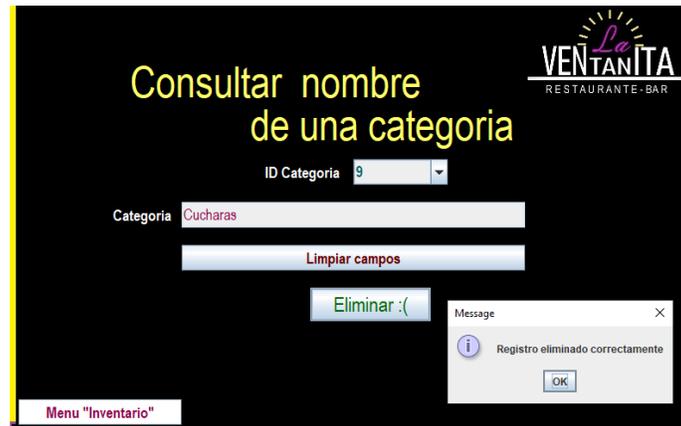


Figura 85. Resultado de prueba CP32_ConEliCat.

Las pruebas para las interfaces de “inventario” se describen en **tablas 43, 44, 45 y 46**. Los resultados de las pruebas realizadas se visualizan en las figuras **86, 87, 88 y 89** respectivamente.

| Caso de prueba | Nuevo producto de inventario |
|------------------------------------|--|
| Identificador caso de prueba | CP33_NueProInv |
| Función a probar | Agregar producto a inventario / materias primas. |
| Objetivo | Validar la función de agregar producto a inventario. |
| Descripción | Se agregarán datos en la interfaz de manera que se guarde un nuevo producto en el registro de inventario. |
| Criterios de éxito | Lista de criterios de éxito: 1. El registro queda almacenado en la base de datos. 2. Se muestra un mensaje que confirma el almacenamiento. |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra una alerta en pantalla de error. 2. El registro no se realiza. |
| Precondiciones | Conocer los datos a almacenar y que no exista un registro previo bajo el mismo nombre. |
| Perfil del usuario | Gerente / Administrador. |
| Necesidades para el caso de prueba | Datos de prueba: fecha: 2020-12-1, Nombre: Chuletas de cerdo, Cantidad: 10, unidad de medida: kilo, categoría: carnes, identificador de usuario: 8 |

| | | | |
|--------------------------|--|---|---|
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario entra al menú "inventario", y selecciona "nuevo producto" en el apartado de "inventario". | Se muestra la interfaz de formulario de "nuevo producto de inventario". |
| | 2 | Entradas: fecha: 2020-12-1, Nombre: Chuletas de cerdo, Cantidad: 10, unidad de medida: kilo, categoría: carnes, identificador de usuario: 8 Descripción: Se llenan los campos con los datos y se presiona el botón de "guardar". | Se muestra un mensaje en pantalla que confirma el registro de la información. |
| Post condiciones | El registro se muestra en el apartado de "ver inventario". | | |

Tabla 43. Caso de prueba 33 Nuevo producto de inventario.



Figura 86. Resultado de prueba CP33_NueProInv.

| | | | |
|------------------------------------|--|--|--|
| Caso de prueba | Modificar datos de producto de inventario / materias primas | | |
| Identificador caso de prueba | CP34_ModProInv | | |
| Función a probar | Interfaz de modificación de datos. | | |
| Objetivo | Modificar un artículo del inventario. | | |
| Descripción | Se modificará un producto introduciendo datos nuevos al registro. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Las modificaciones quedan registradas en el sistema. 2. Se muestra un mensaje en pantalla que confirma la modificación del producto. | | |
| Criterios de falla | Lista de criterios de falla: 1. El registro no queda guardado y se muestra un mensaje de error en pantalla. | | |
| Precondiciones | Conocer el identificador del producto a modificar. | | |
| Perfil del usuario | Gerente / Administrador | | |
| Necesidades para el caso de prueba | Datos de prueba: ID:4 cantidad: 5, fecha: 2020-12-02 | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | 1 | El usuario accede al apartado de modificar producto en el menú de "inventario". | Se muestra la interfaz de "modificar producto de inventario". |
| | 2 | Entradas: ID:4 cantidad: 5, fecha: 2020-12-02 Descripción: Se selecciona la id a modificar y se modifican los campos de texto de cantidad y fecha. Se oprime el botón de "modificar". | Muestra una alerta que confirma que la modificación se ha hecho de manera exitosa. |
| Post condiciones | Al seleccionar de nuevo la id modificada se deben visualizar los datos nuevos | | |

Tabla 44. Caso de prueba 34 Modificar producto de inventario.



Figura 87. Resultado de prueba CP34_ModProInv.

| | | | |
|------------------------------------|--|----------------------------|----------------|
| Caso de prueba | Consulta y eliminación de un producto de inventario | | |
| Identificador caso de prueba | CP35_ConEliProInve | | |
| Función a probar | Buscar y eliminar un producto de inventario almacenado en el sistema. | | |
| Objetivo | Eliminar producto de inventario. | | |
| Descripción | Se eliminará una producto del registro del inventario. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestra un mensaje en pantalla que confirma la eliminación del producto. 2. El ID del producto ya no aparece en el botón de búsqueda. | | |
| Criterios de falla | Lista de criterios de falla 1. Se muestra un mensaje de error en pantalla. | | |
| Precondiciones | El producto debe estar almacenado en los registros del sistema. | | |
| Perfil del usuario | Gerente / Administrador. | | |
| Necesidades para el caso de prueba | Datos de prueba: Id: 4 | | |
| Autor | Miguel Angel Cuaya Tepeyahuil. | | |
| Fecha de creación | 1-12-2020 | | |
| | No paso | Usuario del sistema | Sistema |

| | | | |
|--------------------------|---|--|---|
| Flujo del caso de prueba | 1 | El usuario ingresa al apartado de consulta en el menú de "inventario" en la sección de "inventario". | Muestra la interfaz de consulta y eliminación de productos de inventario. |
| | 2 | Entradas: ID: 4 Descripción: El usuario selecciona la Id del producto que desea borrar. | Se rellena los campos con la información asociada al producto. |
| | 3 | Descripción: Se oprime el botón de "eliminar". | Aparece un mensaje en pantalla que confirma la eliminación del producto. |
| Post condiciones | El id que se eliminó ya no aparece entre las opciones seleccionables. | | |

Tabla 45. Caso de prueba 35 Consulta y eliminación de un producto del inventario.



Figura 88. Resultado de prueba CP35_ConEliProInve.

| Caso de prueba | de | Registro de Inventario |
|------------------------------|----|---|
| Identificador caso de prueba | | CP36_RegInv |
| Función a probar | | Interfaz donde se muestran todos los productos registrados del inventario. |
| Objetivo | | Verificar el correcto funcionamiento de la interfaz. |
| Descripción | | Se ingresará al apartado "registro" de inventario para ver todos las registros de materias primas del inventario. |

| | |
|------------------------------------|--|
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestran todos los productos almacenados en la base de datos hasta ese momento. |
| Criterios de falla | Lista de criterios de falla: 1. No se muestra nada en la interfaz. |
| Precondiciones | Se debe tener registrado al menos un producto de inventario en el sistema. |
| Perfil del usuario | Gerente o Mesero. |
| Necesidades para el caso de prueba | Datos de prueba: Ninguno. |
| Autor | Miguel Angel Cuaya Tepeyahuitl. |
| Fecha de creación | 1-12-2020. |
| | No paso Usuario del sistema Sistema |
| Flujo del caso de prueba | |
| | 1 El usuario accede al apartado de "registro" en la sección de inventario. Muestra todos productos que se han dado de alta hasta ese momento. |
| Post condiciones | Se muestra la fecha de modificación, nombre del producto, cantidad, unidad de medida, categoría, identificador del gerente que realizo la última modificación y su nombre. |

Tabla 46. Caso de prueba 36 Registro inventario.

| Fecha: | Nombre: | Cantidad: | Unidad: | Categoría: | ID gerente: | Quien realizo: |
|------------|-----------------|-----------|---------|------------|-------------|--------------------|
| 2020-08-29 | Aceite de oliva | 4 | Botella | Abarrotos | 7 | Jacqueline Ramí... |
| 2020-11-01 | Aarochera | 3 | Kilo | Carne | 10 | Dayana Quinter... |

Menu "Inventario"

Figura 89. Resultado de prueba CP36_RegInv.

Las pruebas para las interfaces de “mesero” se describen en **tablas 47 y 48**. Los resultados de las pruebas realizadas se visualizan en las **figuras 90, 91 y 92** respectivamente. Nota: Las figuras 91 y 92 pertenecen a un caso de prueba.

| | | | |
|------------------------------------|---|---|---|
| Caso de prueba | Nuevo registro de orden | | |
| Identificador caso de prueba | CP37_NueRegOrd | | |
| Función a probar | Interfaz de registro de una nueva orden. | | |
| Objetivo | Comprobar el funcionamiento de la interfaz de registro de nueva orden. | | |
| Descripción | Se comprobará el funcionamiento de la interfaz “nuevo registro de orden”. | | |
| Criterios de éxito | Lista de criterios de éxito: 1. El registro se hizo correctamente. 2. Se muestra una alerta en pantalla que confirma el registro. | | |
| Criterios de falla | Lista de criterios de falla: 1. Aparece en pantalla un mensaje de error. | | |
| Precondiciones | Se tiene que ser forzosamente un usuario de tipo mesero. | | |
| Perfil del usuario | Mesero | | |
| Necesidades para el caso de prueba | Datos de prueba: fecha 2020-01-23, id:6, nota: Mucha sal a la comida. | | |
| Autor | Miguel Angel Cuaya Tepeyahuitl | | |
| Fecha de creación | 1-12-2020 | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | | | |
| | 1 | El usuario tipo mesero selecciona la opción de “nuevo registro de orden” | Se muestra la interfaz de “nuevo registro de orden” |
| | 2 | Entradas: fecha 2020-01-23, id: 6, nota: Mucha sal a la comida. Descripción: El usuario selecciona el id del mesero que atenderá la orden, escribe | Se muestra un mensaje que confirma el registro de la nueva orden. |

| | | | |
|------------------|--|--|--|
| | | la fecha y procede a anotar una nota a cocina. Para finalizar se presiona el botón de “guardar”. | |
| Post condiciones | El registro nuevo ahora es seleccionable | | |

Tabla 47. Caso de prueba 37 Nuevo registro de orden.



Figura 90. Resultado de prueba CP37_NueRegOrd.

| | |
|------------------------------------|--|
| Caso de prueba | Agregar platillos a una orden |
| Identificador caso de prueba | CP38_AgrPlaOrd |
| Función a probar | Interfaz que agrega platillos a un registro de orden. |
| Objetivo | Comprobar el funcionamiento de agregar platillos a una orden. |
| Descripción | Se agregarán platillos a una orden existente. |
| Criterios de éxito | Lista de criterios de éxito: 1. Se muestra un mensaje en pantalla que confirma que se ha agregado un platillo a la orden. |
| Criterios de falla | Lista de criterios de falla: 1. Se muestra un mensaje de error en pantalla. |
| Precondiciones | La orden a la que se le desea agregar platillos debe estar registrada. |
| Perfil del usuario | Mesero. |
| Necesidades para el caso de prueba | Datos de prueba: Id:3, producto: 3,5 |

| | | | |
|--------------------------|---|--|---|
| Autor | Miguel Angel Cuaya Tepeyahuitl. | | |
| Fecha de creación | 1-12-2020. | | |
| | No paso | Usuario del sistema | Sistema |
| Flujo del caso de prueba | | | |
| | 1 | El usuario selecciona la opción de “agregar platillo a una orden”. | Se muestra en pantalla la interfaz donde se agrega el platillo. |
| | 2 | Entradas: Id: 3, producto: 3 Descripción: El usuario selecciona el identificador de la orden para posteriormente seleccionar el producto a agregar. Para finalizar se presiona el botón de “agregar”. | Se muestra un mensaje en pantalla que confirma que el platillo se agregó a la orden seleccionada. |
| | 3 | Entradas: Id: 3, producto: 5 Descripción: El usuario selecciona el identificador de la orden para posteriormente seleccionar el producto a agregar. Para finalizar se presiona el botón de “agregar”. | |
| Post condiciones | El platillo agregado es visible en el listado de platillos de la orden. | | |

Tabla 48. Caso de prueba 38 Agregar platillos a una orden.



Figura 91. Resultado de prueba CP38_AgrPlaOrd.



Figura 92. Resultado 2 de prueba CP38_AgrPlaOrd.

4. Conclusiones y recomendaciones

Gracias a la realización de este proyecto se obtuvo una mayor experiencia respecto a realizar un proyecto fuera de la universidad adentrándonos a los problemas reales que tienen las pequeñas empresas que no cuentan con un sistema de control de datos, y como estudiante aprender a utilizar nuevas herramientas, sin olvidar el reforzamiento de conocimientos de diseño de base de datos.

El desarrollo fue complicado, el mayor problema que se presentó fue reconstruir todo el sistema de último momento, empezando por la estructura de la base de datos. Al mostrar un prototipo que tenía en funcionamiento el usuario lo evaluó, como resultado surgieron algunos cambios, lo que provocó hacer cambios en la base de datos y adaptarla al sistema, quitando demasiado tiempo para la realización de la documentación. Otra dificultad fue la realización de la documentación del proyecto, pues tuve necesidad de aprender a redactar en tercera persona sin utilizar demasiados tecnicismos.

Una recomendación para futuros trabajos es: realizar una investigación de requerimientos más amplia y la búsqueda de herramientas que puedan optimizar el desarrollo del proyecto, mantenernos actualizados con los lanzamientos de actualizaciones en programación y sobre todo leer para realizar una correcta documentación.

5. Referencias bibliográficas

[1] URL: <http://www.sidar.org/recur/desdi/traduc/es/visitable/tecnicas/Prototyping.html> Pagina de SID@R, en ella se puede consultar información sobre el modelo prototipado. Fecha de consulta: 23/octubre/2020

[2] Rettig, Marc, "Prototyping for Tiny Fingers (Everything I Need to Know About Prototyping, I Learned In Kindergarten)", Communications of the ACM, Estados Unidos, 1994.

[3] URL: <https://rockcontent.com/es/blog/que-es-java/> Página de rockcontent, en ella se pueden conseguir cursos sobre marketing digital

[4] Arnold, Ken, Gosling, James, Holmes, David, "El lenguaje de programación Java" Editorial Addison Wesley, España, 2001 Disponible en: <https://docs.google.com/file/d/0Byy7aUI9u4fBRnJwc1U5Vkdnlk/edit>

[5] Ladrón de Guevara, Jorge Martínez, "Fundamentos de programación en Java" Editorial EME, España, 2010

[6] The phpMyAdmin devel team, "phpMyAdmin Documentation Versión 4.8.0-dev", 2018 disponible en: <https://www.pdf-manual.es/bases-de-datos/201-phpmyadmin-documentation.html>

[7] Gomez, Enrique, "Desarrollo de software con NetBeans 7.1" Editorial marcombo, España, 2014

[8] MySQL, "MySQL 5.7 Reference Manual", 2016 disponible en:
<http://dev.mysql.com/doc/refman/5.7/en/>

[9] Lawrence, Shari. "Ingeniería del software". Editorial PRENTICE HALL, Argentina, 2002

[10] URL: https://en.wikipedia.org/wiki/Microsoft_Paint Página de Wikipedia (enciclopedia gratuita), en ella se puede encontrar información sobre casi cualquier tema.



Universidad Politécnica de Puebla
Ingeniería en Informática

*Cuaya Tepeyahuitl Miguel Angel
Simbro García Karen Mónica
Rodríguez Huesca Rebeca*

Este documento se distribuye para los términos de la
Licencia 2.5 Creative Commons (CC-BC-NC-ND 2.5 MX)