EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

“APLICACIÓN MULTITÁCTIL MULTIUSUARIO PARA REGISTRAR COMANDAS EN RESTAURANTES”

TESIS

PRESENTADA COMO REQUISITO PARCIAL
PARA OBTENER EL GRADO DE:
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

ANGEL EDUARDO GAXIOLA JAVIER

DIRECTOR:
M.C RAFAEL ARMANDO GALAZ BUSTAMANTE

Hermosillo Sonora, México 25 de Agosto 2021

INSTITUTO TECNOLÓGICO DE HERMOSILLO

ISO 9001:2015
Sistema de Gestión de Calidad Certificado

México 2021 Año de la Independencia

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Hermosillo
División de Estudios de Posgrado e Investigación

SECCIÓN: DIV. EST. POS. E INV.
No. OFICIO: DEPI/170/21
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS.

12 de julio de 2021

C. ANGEL EDUARDO GAXIOLA JAVIER,
P R E S E N T E.

Por este conducto, y en virtud de haber concluido la revisión del trabajo de tesis que lleva por nombre **"Aplicación Multitáctil Multiusuario para registrar Comandas en Restaurantes"**; que presenta para el examen de grado de la MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN, y habiéndola encontrado satisfactoria, nos permitimos comunicarle que se autoriza la impresión del mismo a efecto de que proceda el trámite de obtención de grado.

Deseándole éxito en su vida profesional, quedo de usted.

A T E N T A M E N T E

M.C. RAFAEL ARMANDO GALAZ BUSTAMANTE
DIRECTOR

M.C. MARÍA TRINIDAD SERNA ENCINAS
SECRETARIA

S.E.P.

INSTITUTO TECNOLÓGICO DE HERMOSILLO
DIVISIÓN DE ESTUDIOS DE POSGRADO

M.C. CÉSAR ENRIQUE ROSE GÓMEZ
VOCAL

M.C.O. ROSA IRENE SÁNCHEZ FERMÍN
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

RISF/eme*

Av. Tecnológico S/N Col. El Sahuaro C.P. 83170 Hermosillo, Sonora
Tel. 01 (662) 260 65 00, ext. 136, e-mail: depi_hermosillo@tecnm.mx
tecnm.mx | ith.mx

ISO 9001:2015

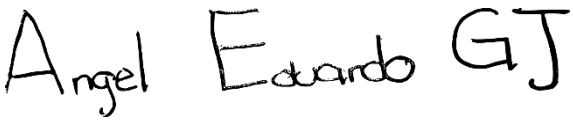
México 2021 Año de la Independencia

CARTA CESIÓN DE DERECHOS

En la ciudad de Hermosillo Sonora a el día 25 de Agosto del año 2021 el que suscribe C. “Angel Eduardo Gaxiola Javier”, alumno de la maestría en “Ciencias de la computación” adscrito a la División de Estudios de Posgrado e Investigación, manifiesta que es autor intelectual del presente trabajo de Tesis titulado “Aplicación Multitáctil Multiusuario para Registrar Comandas en Restaurantes” bajo la dirección de Rafael Armando Galaz Bustamante y ceden los derechos del mismo al Tecnológico Nacional de México/Instituto Tecnológico de Hermosillo, para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben de reproducir el contenido textual, graficas, tablas o datos contenidos sin el permiso expreso del autor y del director del trabajo. Este puede ser obtenido a la dirección de correo electrónico siguiente: “angeleduardogj@gmail.com”. Una vez otorgado el permiso se deberá expresar el agradecimiento correspondiente y citar la fuente del mismo.

ATENTAMENTE



Angel Eduardo Gaxiola Javier

AGRADECIMIENTOS

Es para mí una gran satisfacción poder dedicarles este proyecto a Dios y a cada uno de mis seres queridos, quienes han sido mis pilares para seguir adelante.

Agradezco mucho a mis padres Sandra Javier Saiz y Eduardo Gaxiola Grajeda, porque ellos son la motivación de mi vida, constructores de lo que seré. A mi hermana Alejandra Anahí Gaxiola Javier, porque eres la razón de sentir tanto orgullo, gracias por confiar siempre en mí.

Agradezco a mis compañeros y profesores por compartir conocimiento y por todo el compañerismo. Al Instituto Tecnológico de Hermosillo y al Consejo Nacional de Ciencia y Tecnología (CONACYT) que fueron el camino para llevar a cabo mis estudios de maestría.

Gracias a la vida y a todas las personas que me han abierto las puertas para obtener este nuevo triunfo.

RESUMEN

La industria de alimentos y bebidas en México se encuentra en constante cambio y crecimiento. Para algunos restaurantes es difícil adaptarse, lo que suele ser un problema para que logren obtener los resultados esperados.

Los negocios restauranteros buscan reajustar sus procesos de trabajo con ayuda de herramientas tecnológicas. Esperan ser más competitivos ante otras opciones de consumo, ofreciendo mejores servicios y logrando satisfacer las necesidades de los clientes.

En esta investigación se propone el uso de una aplicación multitáctil multiusuario para registrar comandas en restaurantes, de esta forma automatizar el registro de ordenes en restaurantes. Así un restaurante puede ofrecer el auto servicio en establecimiento lo que logrará disminuir el tiempo y los errores presentados al momento de realizar el registro de la comanda de un comensal.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

Capítulo 1.

1. Introducción.

La industria restaurantera en México se encuentra en constante crecimiento, lo que da origen a cambios y retos. Por esto los restaurantes ofrecen más y mejores servicios, además de buscar convertirse en una buena oferta en la industria. Algunos reajustan sus procesos haciendo uso de distintas tecnologías lo que genera competencia y puede ser más atractivo para los clientes.

Un ejemplo del uso de distintas tecnologías en esta industria, es el registro de la comanda en restaurantes con mesas interactivas y aplicaciones multitáctil multiusuario. Los siguientes antecedentes muestran distintos casos reales de estas tecnologías o similares para ofrecer una buena experiencia al comensal.

1.1. Antecedentes.

El restaurante EL OCHO CAFÉ RECREATIVO en Ciudad de México desde 2009, ofrece a sus comensales una agradable experiencia de entrega de órdenes (Alimentos y bebidas). Con el uso de una mesa interactiva ponen a disposición juegos digitales. El diseño y desarrollo, así como el contenido multimedia fueron realizados e integrados en su totalidad por la empresa de tecnología SIETEMEDIA [1].

La empresa SIETEMEDIA con la colaboración del corporativo DUPONT, implementó en Ciudad de México en el año de 2012 una mesa interactiva para el restaurante MUTO. Utilizó la superficie inteligente CORIAN, para generar en los comensales una experiencia distinta en el registro de órdenes. Lamentablemente este proyecto cerró al público en 2013 [2].

Los antecedentes anteriores muestran la posibilidad de aplicar estas herramientas tecnológicas en restaurantes, describen como pueden mejorar la satisfacción del cliente mejorando la calidad de los servicios en su implementación. Para analizar más a detalle lo siguiente, se realizó el estudio de la siguiente documentación.

Según la ACADEMIA MEXICANA DE COMPUTACIÓN una superficie interactiva tiene la capacidad de ofrecer contenido multisensorial (Audio, imágenes y videos) a los usuarios. La revista académica INGENIARE retoma la definición añadiendo el término de interacciones basadas en gestos, lo define como parte de sistemas interactivos, que permiten la manipulación de elementos digitales de una manera más fácil para el humano [3,4].

DINE 'N' FINE TABLE es un proyecto publicado en el año de 2018 en la conferencia internacional SMART CITY AND EMERGING TECHOLOGY (ICSCET). Con ayuda de interfaces humano-computadora que permiten la facilidad de la interacción, así como reducir el tiempo y el esfuerzo que se pueda tomar para el uso de estas plataformas. El objetivo para el proyecto de investigación es el mostrar las posibilidades de la tecnología multitáctil, ofreciéndola como una interfaz humano-computadora que permite mejorar la experiencia de los clientes. En el proyecto implementan herramientas que permite a un restaurante ofrecer sus servicios en una mesa interactiva. Con el uso de una cámara y algún dispositivo táctil, han logrado determinar una exactitud y fiabilidad para la manipulación de elementos virtuales obteniendo como resultado un rendimiento óptimo, determinando la fiabilidad de estos equipos [5].

La revista digital DELOITTE INSIGHTS publicada en el año de 2019 por la empresa de servicios tecnológicos DELOITTE, toma como titular "Tendencias tecnológicas del 2019 más allá de la frontera digital". En esta publicación se redacta un artículo que lleva de nombre "Interfaces inteligentes re-imaginando la manera como humanos, máquinas y datos interactúan", donde describe este tipo de interfaces humano-computadora cómo interfaces que combinan lo último en técnicas de diseño centrado en lo humano y además de distintas técnicas que transforman las tradicionales interfaces humano-computadora, generando nuevas formas de interacción entre máquinas, datos y humanos. Para lograr esto se debe de combinar un conjunto de tecnologías como hardware con pantallas táctiles, cámaras, micrófonos, y distintos sensores por ejemplo biométricos. También software que permite procesar los datos obtenidos como comandos de voz, procesamiento de imágenes, etc. Se puede integrar en su conjunto para desarrollar

productos que cuentan con un diseño de interfaces inteligentes, en los cuales es necesario aplicar otras tecnologías como lo son realidad virtual, la realidad aumentada, el internet de las cosas, tecnologías en la nube y computación de frontera que son tecnologías que permiten añadir complejidad a la experiencia que se puede ofrecer al usuario [6].

1.2. Planteamiento del problema a solucionar.

Según el censo económico contemplado hasta 2014 del INEGI (Instituto Nacional de Estadística y Geografía) en el año 2013 los establecimientos dedicados a la preparación de alimentos y bebidas representaron el 1.1% del PIB (Producto Interno Bruto) mexicano. Lamentablemente en los últimos diez años la tendencia se mantiene a la baja, donde cifras del año 2003 reflejaron una contribución de 1.7% al PIB mexicano [7].

En 2018 la revista bajo la licencia de NEWSWEEK del GRUPO EDITORIAL CRITERIO en la versión de NEWSWEEK MÉXICO, publicó una entrevista realizada a Francisco Fernández Alonso presidente de CANIRAC (CÁMARA NACIONAL DE LA INDUSTRIA DE RESTAURANTES Y ALIMENTOS CONDIMENTADOS) en México, la cual fue republicada por la AMR (ASOCIACIÓN MEXICANA DE RESTAURANTES). Se relata cómo la industria de preparación de alimentos y bebidas se enfrenta a distintos problemas y retos. Uno de ellos es la gran y diversa oferta que existen de restaurantes en México, donde el reto para ser más competitivo es entender qué es lo que busca el cliente. La solución apunta a la experiencia que un restaurante puede ofrecer a sus clientes en sus servicios, para ello la adaptación a los cambios de la industria es importante para que un restaurante pueda ser rentable [8,9].

La investigación FOODY-SMART RESTAURANT MANAGMENT AND ORDERING SYSTEM describe cómo además de la calidad de los alimentos y bebidas, la atención que un restaurante puede ofrecer a sus comensales es determinante para definir su reputación y mantener una base de clientes [10].

El tiempo y la precisión para registrar una comanda son factores claves que influyen en la evaluación de un restaurante. Siendo estas dos, deficiencias

comunes. Por ejemplo, la alta cantidad de tiempo de espera en una larga fila que un comensal puede pasar para realizar un pedido, además del momento en que un cajero o mesero hace un registro de pedidos con errores [10].

Ante esta información, se genera la inquietud de ofrecer una estrategia para que los restaurantes puedan ofrecer un mejor servicio a sus comensales. Con el uso de una aplicación multitáctil multiusuario que puede formar parte de un sistema de registro para comandas en restaurantes, para su implementación en mesas interactivas. Buscando reflejar al instante la interacción del comensal con el servicio y disminuir los errores para registrar comandas en restaurantes.

1.2.1. Preguntas de investigación.

- ¿Qué variables definen la satisfacción del cliente al realizar un pedido?
- ¿Cuál es la arquitectura de software a emplear?
- ¿Qué tecnologías son útiles para desarrollar una aplicación multitáctil multiusuario?
- ¿Cuál es el diseño de interfaz de usuario ideal para una aplicación multitáctil multiusuario para el registro de comandas en restaurantes?

1.2.2. Pregunta de investigación principal.

- ¿Es posible desarrollar una aplicación multitáctil multiusuario para mesas interactivas que pueda ser integrado a un sistema de registro de comandas en restaurantes?

1.3. Objetivos.

Para el desarrollo de la aplicación se han planteado los siguientes objetivos.

1.3.1. Objetivo general.

Desarrollar una aplicación multitáctil multiusuario para mesas interactivas que pueda ser integrado a un sistema de registro de comandas en restaurantes.

1.3.2. Objetivos específicos.

- Proponer una arquitectura de software para una aplicación multitáctil multiusuario para registrar comandas en restaurantes.
- Diseñar una interfaz de usuario ideal para la experiencia multitáctil multiusuario.
- Diseñar una aplicación multitáctil multiusuario para disminuir el tiempo de espera y aumentar la precisión al momento de registrar comandas en restaurantes.
- Desarrollar una aplicación multitáctil multiusuario para registrar comandas en restaurantes según se haya diseñado.
- Integrar la aplicación multitáctil multiusuario desarrollada a un sistema de registro de comandas en restaurantes.

1.4. Alcances y delimitaciones.

La aplicación debe ser integrada a un sistema de registro de comandas: La aplicación es una interfaz que permite al usuario interactuar con el servicio de pedidos de alimentos y bebidas que ofrece un restaurante. Esta aplicación funciona como FRONTEND que debe consumir una API REST de un sistema de registro de comandas.

La propuesta no incluye un sistema de registro de comandas: Esta propuesta no incluye un sistema de registro de comandas, que tiene que utilizar la aplicación multitáctil multiusuario. Este sistema se simula con una API REST en la implementación.

La aplicación permite la interacción a varios usuarios a la vez: Parte del objetivo es desarrollar una aplicación multitáctil multiusuario, donde varios usuarios sean capaz de interactuar sin interferir uno con el otro. Esta implementación se validó para cuatro usuarios, debido a límites técnicos.

1.5. Justificación.

Según la publicación GESTIÓN TECNOLÓGICA EN RESTAURANTES: DESARROLLO Y VALIDACIÓN DE UN INSTRUMENTO DE MEDICIÓN la importancia del uso de herramientas tecnológicas ha incrementado en la industria restaurantera, ya que al ser usadas representan una oportunidad ante una industria con elevados niveles de incertidumbre [11].

El análisis de implementaciones tecnológicas en un restaurante hecho por JOURNAL OF HOSPITALITY AND TOURISM TECHNOLOGY, afirma que el uso de la tecnología es un factor importante en las operaciones comerciales, esto es consecuencia de los avances tecnológicos [12].

La revista RESTAURANT TECHNOLOGY STUDY en su entrega anual del 2018 publicó cómo los restaurantes se han vuelto en los últimos años cada vez más digitales y basados en datos, debido al compromiso de mejorar sus capacidades con el uso de analíticas [13].

Un restaurante que busca la mejora de sus actividades tiene como consecuencia mejores productos y servicios. Esto al incluir hardware y software en el núcleo de los procesos de las actividades restauranteras, acompañado con el manejo de la información así logrando un negocio centralizado en el cliente.

De ahí que resulte relevantes trabajos de investigación que propongan la implementación de tecnologías en restaurantes, capaces de mejorar los productos y servicios. Razón de esta investigación que busca el desarrollo de una aplicación multitáctil multiusuario para el registro de comandas en restaurantes.

1.6. Aportación del trabajo.

Este trabajo de investigación tiene como finalidad el ser una herramienta para el lector interesado en el desarrollo de aplicaciones multitáctil multiusuario, además de restaurantes en México que busquen implementar esta tecnología como parte de sus procesos.

1.7. Metodología.

Fase 1: Buscar, analizar y recopilar documentación para conocer a detalle la problemática y sustentar la propuesta de solución.

Fase 2: Analizar y diseñar la aplicación a desarrollar, mediante diagramas utilizando el lenguaje unificado de modelado (Diagramas UML).

Fase 3: Desarrollar la aplicación multitáctil multiusuario para el registro de comandas en restaurantes.

Fase 4: Analizar y documentar los resultados obtenidos.

1.8. Organización de la tesis.

Capítulo 2: Estado del arte, presenta el análisis de la documentación de diferentes fuentes que inciden directamente con la problemática a resolver.

Capítulo 3: Análisis y diseño, describe la aplicación a desarrollar mediante algunos diagramas UML.

Capítulo 4: Implantación, documenta el desarrollo de la aplicación multitáctil multiusuario y su integración a un sistema mediante una API REST.

Capítulo 5: Análisis de resultados, muestra el funcionamiento de la aplicación multitáctil multiusuario para que un comensal pueda registrar comandas en restaurantes junto a un sistema de registro de comandas para restaurantes.

Capítulo 6: Conclusiones y trabajo futuro, resume el trabajo de investigación realizado.

Capítulo 2.

2. Estado del arte.

2.1. Introducción.

El desarrollo de tecnología multitáctil involucra un análisis previo a su aplicación, ya que esta tecnología es bastante costosa en consumo de recursos de hardware y económicos. Si se usa sin una planificación podría no generar el rendimiento esperado. Esta es la razón por la cual se tiene que desglosar en pequeños componentes y se tiene que conocer cuáles de ellos pueden ser de utilidad. Además de conocer la manera en que estos se aplican o integran con otros tipos de tecnologías.

2.2. Comanda.

La comanda es un instrumento físico, normalmente presentado en papel. La cual dispone cada persona responsable en tomar pedidos a los clientes. Esta es una parte crítica en el servicio de alimentos y bebidas, ya que es el registro de todo aquello que un cliente espera obtener de la oferta de un local gastronómico [14].

Este documento influye en distintos procesos de un restaurante como la administración de la cocina, el control de facturas, en el control de inventarios entre otros [14].

2.3. Mesa interactiva.

Es una pieza de mobiliario adaptado con tecnología moderna que se puede integrar en distintos tipos de ambiente. Una mesa interactiva se complementa de un software para gestionar la información de forma sencilla e intuitiva con el uso de una interfaz gráfica de usuario [15].

Con esta tecnología varios usuarios pueden interactuar al mismo tiempo con información utilizando sus manos [15].

Una mesa interactiva es una solución ideal para hoteles, centros comerciales, hospitales, museos, exposiciones, restaurantes, tiendas y más. [15]

2.4. Pantalla táctil (Touchscreen).

Una pantalla táctil ofrece al usuario usar sus dedos para manipular distintos elementos en pantalla, es una tecnología intuitiva y permite la exploración entre una gran cantidad de información. Podemos encontrar pantallas que permiten el uso de un usuario o múltiples usuarios [16].

Una pantalla táctil puede ser utilizada para distintas actividades, ya que esta tecnología se encuentra en constante desarrollo. Al usar esta tecnología en un dispositivo se podrá observar una buena respuesta, gran precisión al tacto, claridad de pantalla entre otras buenas cualidades [16].

2.5. Tecnologías interactivas.

2.5.1. Tecnología táctil (Touch).

Esta tecnología hace uso de sensores táctiles, normalmente son módulos que se integran a una pantalla. Reacciona mediante un toque directo sobre su superficie lo que funciona como una entrada de datos que puede representar distintas órdenes para un dispositivo y este último muestra un resultado en pantalla que podrá interpretar el usuario [17].

Varias empresas adoptaron esta tecnología para los usuarios finales. Por ejemplo, un teléfono inteligente (Smartphone) es probable que implemente una pantalla con esta tecnología [18].

2.5.2. Tecnología multitáctil (Multi-Touch).

La tecnología multitáctil funciona con el uso interfaces como lo son pantallas táctiles, que podemos encontrar en dispositivos móviles. Permite a los usuarios realizar distintas acciones, por ejemplo, el seleccionar un elemento, desplazarse entre el contenido del dispositivo, acercar o alejar la visión que el dispositivo ofrece entre otras acciones que tengan como fin interactuar con elementos virtuales [19].

2.5.3. Tecnología multitáctil multiusuario (Multi-User Multi-Touch).

Con el beneficio de la tecnología multitáctil y el uso de software moderno se puede lograr que varias actividades puedan ser realizadas de manera simple e intuitiva. Además de que varios usuarios puedan colaborar en el mismo contenido, trabajando uno del lado del otro de manera eficiente sin interrumpir la experiencia entre sí [16].

Un ejemplo de esta tecnología es, cuando se abre simultáneamente diferentes ventanas llamadas “ventanas emergentes”, para crear una experiencia multifacética que fomenta la interacción. Se debe tomar en cuenta distintos factores como la accesibilidad (UI), la usabilidad (UX) y la durabilidad [16].

La tecnología multitáctil multiusuario aplicada a grandes pantallas genera una experiencia donde lo digital se vuelve tangible. Esto permite a las empresas involucrar a sus clientes en sus procesos y exponerlos a información de interés, de esta manera es como se puede retener al cliente [16].

Si el cliente se encuentra en el establecimiento y un empleado desea interactuar con él para conducir a una mayor retención, al utilizar un dispositivo para esta acción, el dispositivo entra en un estado según el nivel de interactividad que el dispositivo permite a los involucrados [16].

Dispositivo en estado pasivo: Si se utiliza un dispositivo que no es táctil, se dice que el dispositivo se encuentra en estado pasivo [16].

Figura 2.1. Dispositivo en estado pasivo [16].

Dispositivo en estado semiactivo: Si el dispositivo es táctil y en él las personas que se encuentran pueden interactuar con este dispositivo secuencialmente, se puede considerar que el dispositivo se encuentra en estado semiactivo [16].

Figura 2.2. Dispositivo en estado semiactivo [16].

Dispositivo en estado activo: Si el dispositivo es táctil y en él las personas que se encuentran pueden interactuar con este dispositivo simultáneamente, se puede considerar que el dispositivo se encuentra en estado activo [16].

Figura 2.3. Dispositivo en estado activo [16].

2.5.4. El uso de tecnología multitáctil multiusuario.

Existen gran versatilidad en el uso de las tecnologías multitáctiles multiusuario, gracias a que los dispositivos que ofrecen esta tecnología se presentan en gran variedad de tamaños y se pueden utilizar en distintas orientaciones [16].

También es importante planificar la dimensión de las pantallas a utilizar y el tipo de audiencia, es clave para una implementación exitosa. Además de las restricciones físicas, se debe considerar no sobrecargar una interfaz al tener demasiados usuarios interactuando al mismo tiempo [16].

Podemos encontrar la tecnología multitáctil multiusuario en quiscos, mesas, pantallas verticales y estos son los requerimientos ideales [16]:

Quioscos:

- Talla: 22 a 55 pulgadas.
- Número de usuarios a interactuar: Recomendado de 1 a 2.
- Ubicaciones: Punto de venta, ferias, mostradores, centros comerciales, hoteles, museos.
- Casos de uso: Señalización digital interactiva, conserje virtual, búsqueda de caminos, catalogo digital.
- Ventajas: Llamativo, promueve el autoservicio, ergonómico.

Figura 2.4. Quioscos [16].

Mesas:

- Talla: 32 a 84 pulgadas
- Número de usuarios a interactuar: Recomendado de 1 a 8
- Ubicaciones: Punto de venta, showrooms, ferias, salas de reuniones, museos.
- Casos de uso: Lluvia de ideas, presentación de productos, discurso de ventas, juegos multijugador, entretenimiento.
- Ventajas: Colaborativo, intimo, ergonómico, versátil.

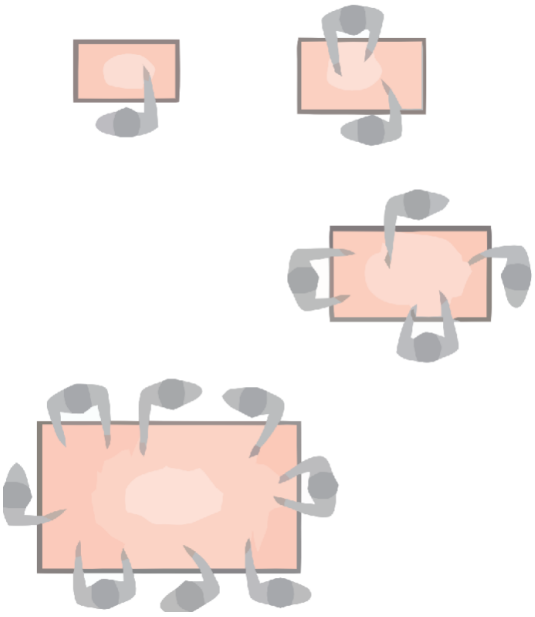


Figura 2.5. Mesas [16].

Pantallas verticales:

- Talla: 32 a 84 pulgadas.
- Número de usuarios a interactuar: Recomendado de 1 a 2.
- Ubicaciones: Showrooms, ferias, salas de reuniones, puntos de venta, fábricas.
- Casos de uso: Reuniones de equipo, señalización digital interactiva, presentación de productos, catalogo digital, lluvia de ideas.
- Ventajas: Llamativo, promueve el autoservicio, requisito.

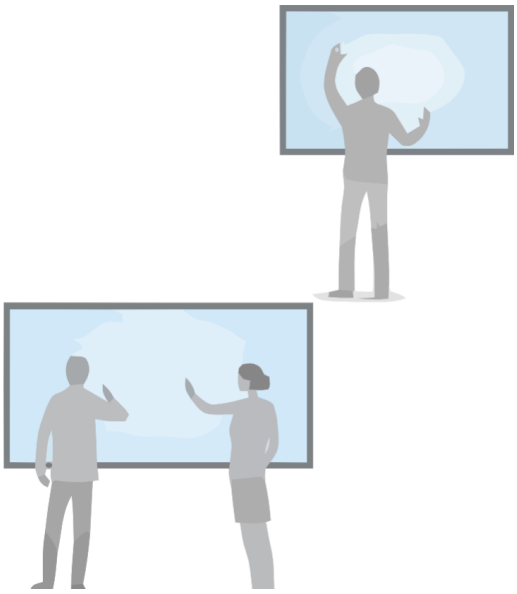


Figura 2.6. Pantallas verticales [16].

2.6. Tipos de tecnología táctil.

Al hablar de tecnología de pantalla táctil, existen diferentes tipos. Cada una con su propio conjunto de características [18].

Son cuatro las más comunes, cada uno tiene diferentes tipos de aplicaciones [18].

2.6.1. Toque resistivo.

La tecnología táctil resistiva se puede encontrar en gran variedad de dispositivos, incluidos monitores táctiles y sistemas de navegación para automóviles [18].

Los paneles resistivos son sensibles a la presión, lo que significa que usan la presión colocada directamente en la pantalla como medio para detectar comandos. Estos paneles táctiles están cubiertos por dos capas que funcionan como electrodos transparentes con separadores entre sí. Como resultado de las dos capas de película transparente en la parte superior del panel, la transmisión de luz no es tan fuerte como en otros tipos de tecnología táctil. El tiempo de vida de este tipo de paneles es corto por qué se debe de aplicar presión en la pantalla, además de ser una tecnología cara [18].

Estos paneles se pueden operar con cualquier tipo de toque por ejemplo al utilizar los dedos, el lápiz óptico e incluso el tacto con guantes [18].

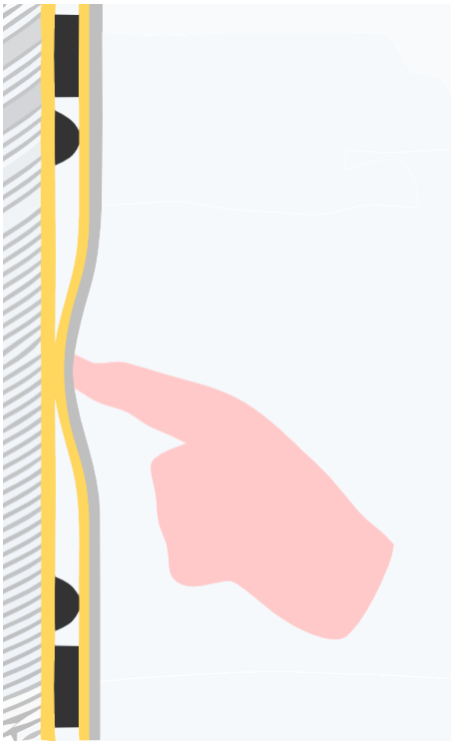


Figura 2.7. Toque resistivo [18].

2.6.2. Toque de imagen óptica.

La tecnología táctil de imágenes ópticas utiliza cámaras infrarrojas y luz para detectar toques ingresados en el panel. La precisión de la detección táctil en las pantallas táctiles de imágenes ópticas puede variar según los componentes utilizados [18].

El reconocimiento táctil de estos dispositivos funciona por medio de imágenes, se puede utilizar cualquier forma de toque por ejemplo al utilizar los dedos, el lápiz óptico e incluso el tacto con guantes [18].

El efecto de la transmisión de luz en las pantallas táctiles de imágenes ópticas tiende a ser muy bueno porque no se utilizan revestimientos obstructores sobre la pantalla. El tiempo de vida estos dispositivos son largo, ya que los toques son ligeros y esto reduce el desgaste [18].

Figura 2.8. Toque de imagen óptica [18].

2.6.3. Toque infrarrojo.

La tecnología táctil infrarroja utiliza la interrupción de algunos haz de luz como un medio para detectar comandos táctiles. En las pantallas táctiles infrarrojas, los haz infrarrojos se organizan en una cuadrícula sobre el panel y los puntos táctiles se calculan cuando los haz de luz se interrumpen [18].

Los paneles táctiles infrarrojos tienden a tener una buena transmisión de luz y son bastantes duraderos. Sin embargo, la experiencia puede ser negativa porque la reacción al tacto puede ser algo lenta [18].

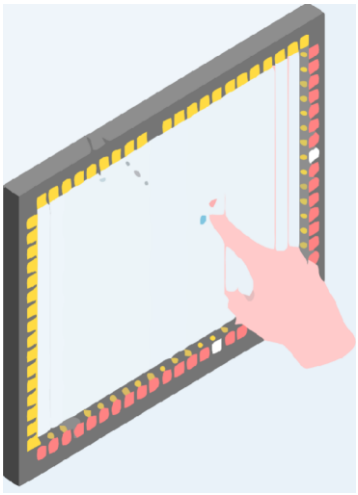


Figura 2.9. Toque infrarrojo [18].

2.6.4. Toque capacitivo.

El toque capacitivo es la tecnología que se usa comúnmente en el mercado de dispositivos táctiles [18].

Esta tecnología comúnmente se utiliza en dispositivos con pequeñas pantallas, por lo que su reconocimiento táctil es de alta precisión y el tiempo de respuesta es muy rápido [18].

Ya que esta tecnología detecta comandos táctiles por medio de corrientes eléctricas, es más fácil de hacer paneles táctiles capacitivos de mayor tamaño.

A diferencia a otras tecnologías, los paneles capacitivos pueden interactúan solo con los dedos o bien con un lápiz conductor [18].

La luz transmitida en los paneles táctiles capacitivos es muy buena, al igual que la precisión de la detección, ya que los recubrimientos de vidrio y plásticos en estas pantallas son generalmente muy duraderos y resistentes al polvo [18].

El toque capacitivo a menudo se promociona cómo la mejor experiencia de usuario de todas las tecnologías táctiles. Principalmente por su gran sofisticación [18].

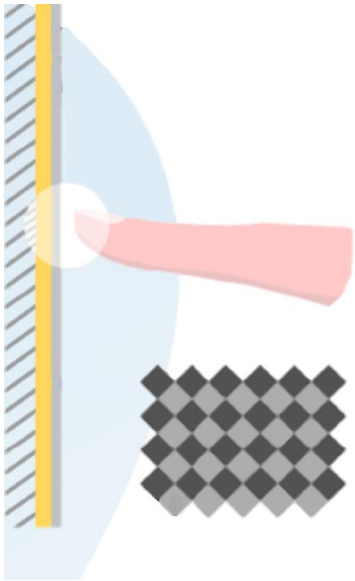


Figura 2.10. Toque capacitivo [18].

2.7. Aplicación multitáctil multiusuario.

Para manipular contenido en los dispositivos con tecnología multitáctil multiusuario, es necesario contar con software preparado para esta finalidad. Para crear una aplicación multitáctil multiusuario, se debe tener en cuenta el tipo de producto o servicio a ofrecer. Existen puntos esenciales que siempre se deben considerar al crear, implementar y mantener una aplicación multitáctil multiusuario [16].

- El software debe ser fluido, reaccionar de manera inmediata a la interacción con el usuario [16].
- Desarrollar el software enfocado en el usuario objetivo, no solo en el contenido. Tomar en cuenta el desarrollo de interfaces gráficas de usuario (UI) y en la experiencia de usuario (UX) [16].
- Se debe considerar de qué manera se utilizará el dispositivo, diseñar el software en consecuencia de como estará montada la pantalla ya sea en un quiosco, mesa o pantalla vertical [16].
- Es ideal desarrollar tecnología nativa, ya que estos ofrecen mejores herramientas que permiten trabajar con la tecnología táctil [16].
- Considerar integrar la aplicación a un sistema de TI existente. Con diferentes herramientas para organizar y administrar datos como CRM, ERP, bases de datos y soluciones de comercio electrónico [16].
- Ser independiente de una conexión a internet, se debe de contar con una solución que sea accesible sin internet y al contar con una conexión toda la información manipulada se debe sincronizar con nuestros servicios en red [16].
- Con la gran cantidad de información recopilada por las aplicaciones implementadas en diferentes dispositivos y en distintas ubicaciones, se debe tener los medios para obtener conclusiones atreves del análisis [16].

2.8. Dispositivos inteligentes.

Un dispositivo inteligente es aquel objeto que cumple con potencia computacional y tiene conexión a internet para compartir datos en la red [20].

Un dispositivo inteligente puede ser un pequeño dispositivo móvil como un teléfono o bien puede llegar a ser un electrodoméstico. Siempre que este objeto este compuesto de componentes que permita recopilar datos físicos como la interacción humana, recopilación de magnitudes físicas y adicional a esto el dispositivo pueda analizar distintos datos [20].

Un dispositivo inteligente está diseñado para interactuar con un usuario como con otro dispositivo que también esté conectado a una red, algunos de estos dispositivos podrían no requerir interactuar directamente con un usuario [21].

2.9. Interfaz humano-computadora.

Existen sistemas compuestos por humanos y máquinas con la finalidad de cumplir distintas actividades gracias a la interacción de estos dos elementos. Cuando se refiere a los humanos se refiere a los operadores de estas máquinas. Las máquinas se refieren a los equipos, herramientas y entornos de trabajo [22].

Por lo que un sistema humano-máquina se conoce como una interfaz entre humanos y máquinas. Estas interfaces permiten la comunicación entre estos dos agentes en forma de paneles donde se puede visualizar y dar seguimiento a las actividades a realizar con el uso de signos, figuras, colores y gráficos que permiten la interacción mediante preguntas y respuestas en formas de menús, tablas, códigos, imágenes, audio, etc. Esta tecnología se conoce como interfaz de usuario [22].

2.9.1. Interfaz inteligente.

El estándar de los siguientes años de las interfaces humano-computadora son establecidas por tecnologías que ofrecen reconocimiento de imágenes, procesamiento de voz, distintos tipos de sensores, inteligencia artificial y tecnologías táctiles [23].

Las interfaces han pasado de entender instrucciones con el uso de botones físicos a procesar gestos y expresiones humanas. El uso de estas tecnologías es cada vez más habitual encontrarlos en distintos dispositivos. Por ejemplo, en teléfonos, computadoras, relojes, asistentes virtuales, autos, refrigeradores por mencionar algunos [23].

2.10. Internet de las cosas.

Es una tecnología que permite que gran cantidad de dispositivos estén conectados a internet. Esta tecnología no solo permite a las personas interactuar con una máquina, además una máquina es capaz de mandar información continuamente a otra máquina [24].

La información recopilada al usar estas redes ayuda a las empresas a conocer mucho mejor a sus clientes, esto es reflejado como valor añadido en sus servicios y productos [24].

2.11. UNITY.

Es un motor gráfico que cuenta con una plataforma de desarrollo que permite crear experiencias interactivas y envolventes. Esta plataforma se encuentra disponible para trabajar en equipos de cómputo con WINDOWS, MAC y LINUX [25].

El manual de usuario de UNITY muestra secciones de trabajo con entornos 2D y 3D. Aspectos visuales como gráficos, físicas y animaciones. Además de manipular elementos de audio [26].

Al desarrollar en esta plataforma se puede encontrar distintas tecnologías que nos permite trabajar distintos proyectos. Para utilizar estas tecnologías, se pide como requisito tener habilidades y conocimiento de programación, bases de datos y redes [26].

2.12. Servicios web.

2.12.1. API.

Un API es una interfaz para aplicaciones de software, esta interfaz da acceso a una aplicación que tiene un conjunto de subrutinas, funciones y procedimientos. De esta manera podemos interactuar con un sistema y sus especificaciones que se compone de una serie de reglas que definen como se interactúa con ese sistema y como se utiliza [27].

2.12.2. API REST.

Es una interfaz entre sistemas que utilizan el protocolo HTTP para ejecutar acciones sobre datos, utilizando distintos formatos como XML y JSON en una red como puede ser internet. Es una alternativa a protocolos de intercambio de comunicación como SOAP (SIMPLE OBJECT ACCESS PROTOCOL) [28].

Cada transferencia por el protocolo HTTP contiene información que será ejecutada por un sistema. Lo que permite que ni el cliente ni servidor necesiten entrar en un estado adicional a la comunicación que se establece entre estos dos elementos [28].

Las operaciones a ejecutar sobre los datos en un sistema donde se involucra REST son cuatro [28]:

- POST: Permite crear recursos.
- GET: Permite consultar recursos.
- PUT: Permite actualizar recursos.
- DELETE: Permite eliminar recursos.

Se puede acceder a un servicio REST utilizando objetos REST. Un objeto REST siempre se manipula a partir de la URI [28].

2.12.3. URI.

La URI es el identificador único de cada recurso de un sistema REST, lo que facilita acceder a la información para operar. Esto facilita la existencia de una interfaz uniforme [28].

Además, una API REST permite realizar operaciones CRUD sobre una base de datos utilizando el protocolo de comunicación HTTP utilizando las URI [29].

2.12.4. CRUD.

El acrónimo CRUD corresponde a las operaciones CREATE, READ, UPDATE y DELETE. Cuyas acciones manipulan una colección de datos que se encuentra en una base de datos [29].

2.12.5. Protocolo HTTP.

HYPERTEXT TRANSFER PROTOCOL o también conocido con sus siglas HTTP, es un protocolo que permite realizar una petición de datos y recursos en la web [30].

Este protocolo fue diseñado en 1990, ha evolucionado con el tiempo. Al analizar el MODELO TCP/IP en la capa de aplicación encontraremos un conjunto de protocolos en donde se encuentra HTTP [30].

2.13. Cliente-Servidor.

Clientes y servidores se comunican intercambiando mensajes individuales con el uso del protocolo HTTP en una red privada o pública, donde los mensajes que envía el cliente se llaman peticiones y los mensajes que envía el servidor se llaman respuestas [30].

2.13.1. Cliente.

Es cualquier herramienta que actúe en representación del usuario [30].

2.13.2. Servidor.

Sirve para responder con recursos según lo solicitado por un cliente. Un servidor puede ser una única entidad o puede estar formado por varios elementos que se reparten las cargas de peticiones [30].

2.14. Base de datos.

Es una tecnología que almacena información y datos de manera organizada. En una base de datos esa información se puede guardar, eliminar, consultar o modificar. Existen distintos tipos de base de datos por ejemplo relacionales y no relacionales [31].

2.15. Trabajos relacionados.

Dine ‘n’ fine table: El objetivo principal del proyecto de investigación es mejorar la experiencia gastronómica del cliente y proporcionar una interfaz agradable y fácil de usar, reduciendo los tiempos y esfuerzos de interacción del cliente. El cliente puede interactuar con dine ‘n’ fine table y así sucesivamente el cliente puede hacer un pedido al chef. Este proyecto está desarrollado con tecnología web y limita la interacción a un usuario [5].

Foody-smart restaurant managment and ordering system: Este trabajo de investigación propone desarrollar un sistema de gestión de restaurantes totalmente automatizado para evitar confusiones entre pedidos, brindar un mejor panorama de la oferta en el servicio de menú y permitir al comensal elegir los productos que ahí se ofrecen, esto sin ser muy específico con el uso de dispositivos móviles [10].

Capítulo 3.

3. Análisis y diseño de la aplicación.

3.1. Introducción.

El siguiente capítulo describe el proceso de análisis y diseño de la aplicación para el registro de comandas en restaurantes y a su vez como éste podrá ser integrado a un sistema externo con el uso de una interfaz.

3.2. Análisis

La captura de los requerimientos tiene como función principal entender lo que el sistema necesita realizar y lo que un usuario espera que haga el sistema. Esto ayuda a realizar el análisis de lo que se necesita para el desarrollo.

3.2.1. Requerimientos funcionales.

Los requerimientos funcionales describen lo que una aplicación o sistema debe de hacer. En la siguiente tabla se presenta los requerimientos funcionales de la aplicación propuesta.

Requerimiento funcional.	Función.	Rol.
RF1	La aplicación debe permitir al comensal abrir una comanda.	Comensal.
RF2	La aplicación permitirá al comensal realizar pedidos en la comanda.	Comensal.
RF3	La aplicación mostrará al comensal las categorías de los productos a ofrecer.	Comensal.
RF4	La aplicación mostrará al comensal los productos que se encuentran dentro de la categoría seleccionada. Si existen subcategorías para esa	Comensal.

	categoría también se mostrará.	
RF5	La aplicación mostrará al comensal los productos que se encuentran dentro de la subcategoría seleccionada.	Comensal.
RF6	La aplicación registrará los productos seleccionados por el comensal. Este conjunto de productos formará un pedido.	Comensal.
RF7	Un comensal podrá modificar su pedido agregando y eliminados elementos de la lista de los productos agregados.	Comensal.
RF8	La aplicación mandará el pedido para que sea preparado.	Comensal.
RF9	La aplicación permitirá al comensal realizar varios pedidos, los cuales se registrarán en una comanda, donde se va registrando el total del consumo.	Comensal.
RF10	La aplicación permitirá al comensal finalizar su consumo registrando la comanda generada y así finalizar su consumo.	Comensal.

Figura 3.1. Requerimientos funcionales.

3.2.2. Requerimientos no funcionales.

Los requerimientos no funcionales representan características generales y restricciones de una aplicación o un sistema. En la siguiente tabla se presenta los requerimientos no funcionales de la aplicación propuesta.

Requerimiento no funcional	Función.
RNF1	La aplicación debe ser multitáctil, podrá interpretar distintos comandos con gestos realizados con dedos. Esto

	atreves de la pantalla táctil del dispositivo.
RNF2	La aplicación debe ser multiusuario, podrá interactuar con 4 usuarios. Cada uno podrá realizar sus actividades de manera fluida y sin que su experiencia sea interrumpida, con el uso de ventanas emergentes.
RNF3	La aplicación debe desplegar un máximo de 4 ventanas emergentes.
RNF4	Cada ventana emergente debe de contar con un marco, que permitirá a los usuarios el manipularlas.
RNF5	Una ventana emergente podrá rotar en sentido del reloj y contra del reloj.
RNF6	Una ventana emergente podrá escalar su tamaño.
RNF7	Una ventana emergente podrá moverse a cualquier punto de la pantalla.
RNF8	Las acciones de las ventanas emergentes deben ser independientes entre sí. Una ventana emergente no podrá influir de alguna manera las acciones de otra.
RNF9	Dentro de las ventanas emergentes deberá desplegar distintas pantallas que permitirán a los usuarios realizar el registro de una comanda.
RNF10	La aplicación debe ser escalable y adaptable a distintos sistemas externos, por lo que debe de consumir una API REST.

Figura 3.2. Requerimientos no funcionales.

3.2.3. Casos de uso.

Estos describen las interacciones básicas que tendrán los usuarios con el sistema. Se representa con el lenguaje de modelado unificado (UML) utilizando el diagrama de casos de uso.

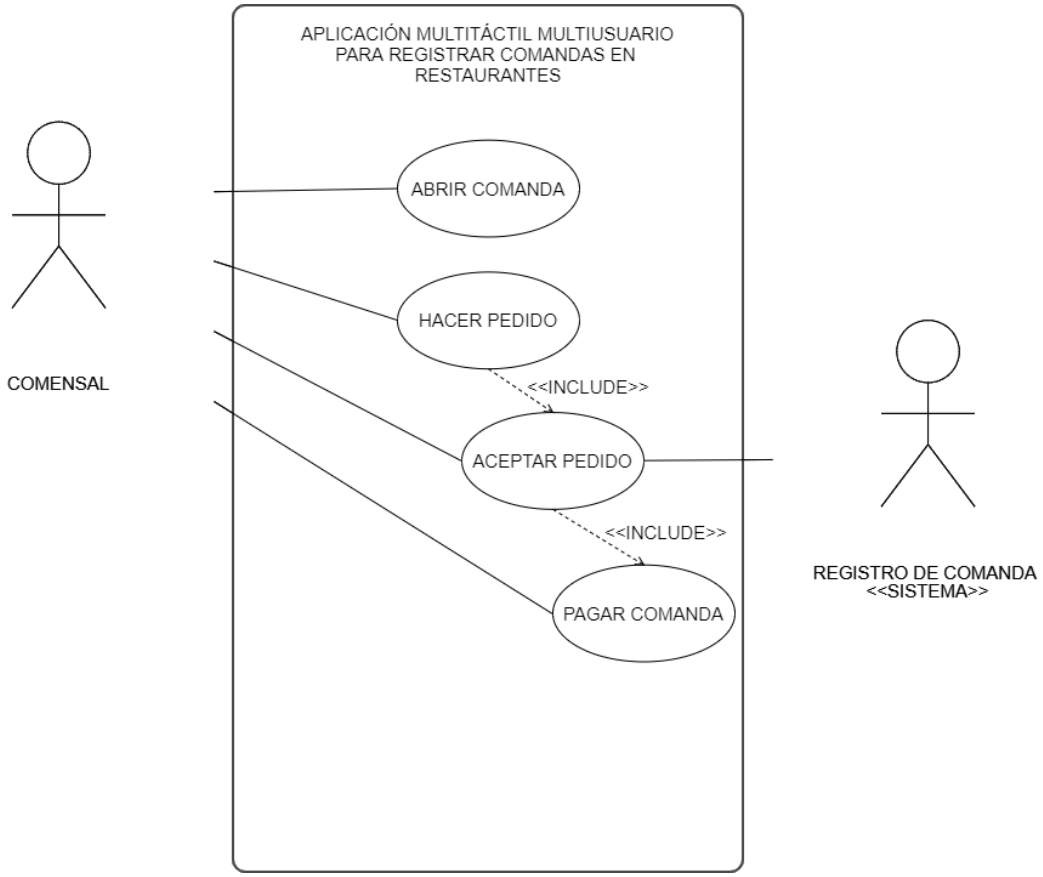


Figura 3.3. Diagrama de casos de uso. Cuenta con 2 actores, el comensal como usuario y un sistema externo de registro de comandas, los cuales interactúan al realizar los pedidos.

3.3. Diseño

3.3.1. Servidor local.

Para el desarrollo de la aplicación es recomendable utilizar un servidor local. El servidor local puede contar con distintas especificaciones de hardware y de software. Para desarrollar la propuesta de la arquitectura del sistema se utilizarán las siguientes especificaciones técnicas para:

Hardware:

- Procesador: INTEL CORE I5-8300H
- RAM: 8.00 GB
- GPU: 4 GB

Software:

- Sistema operativo: WINDOWS 10

Se puede utilizar un equipo personal de trabajo, pero es indispensable contar con estas propiedades para tener un buen rendimiento en el desarrollo, ya que se acercan a los requerimientos técnicos aceptables de distintas herramientas utilizadas. Es importante recordar que las herramientas se actualizan, entonces se tiene que tomar en cuenta que las especificaciones técnicas del servidor fueron elegidas para las versiones de las herramientas aquí utilizadas.

Para el desarrollo se puede optar por un servidor en la nube, en caso de ser así es recomendable utilizar un servidor destinado para desarrollo.

El servidor realizará la función de un sistema de registro de comandas en restaurantes, servicios que consumirá nuestra aplicación multitáctil multiusuario y es importante contar con estos para poder realizar una implementación. Pero el enfoque de este trabajo de investigación es desarrollar la aplicación e integrarla a un servicio ya existente. Al desarrollar los componentes del servidor puede servir para generar una perspectiva del grado de compatibilidad que tendría la aplicación con un sistema de registro de comandas ya existentes en el mercado.

3.3.2. Servidor de bases de datos.

Se utilizará MARIADB como sistema gestor de base de bases de datos, el cual tiene una alta compatibilidad con MYSQL, ya que poseen las mismas APIS y bibliotecas.

3.3.3. Servidor web.

Un servidor web es un software que debe ser instalado en un equipo para que pueda funcionar, en este caso en un servidor local. Este servidor web será desarrollado en PYTHON utilizando un microframework de nombre FLASK. Se utilizan estas tecnologías por la facilidad de configuración y trae consigo una ventaja en tiempo de desarrollo gracias a los módulos integrados de ambas tecnologías.

3.3.4. Dispositivo táctil.

La aplicación multitáctil multiusuario está diseñada para ser implementada en mesas interactivas. Existen distintas opciones de estos dispositivos y en su mayoría son de un costo económico elevado por lo que en este proyecto se simulará con el uso de una tableta. La manera en la que una tableta funciona es muy similar a una mesa interactiva, ya que cuenta con un margen de trabajo amplio (Para una persona) y son táctiles, la limitante es que al usar este dispositivo móvil es viable que sea usada por una persona, ya que entra en un estado semiactivo. Sin embargo, si interactuamos en el área de trabajo solamente con los dedos sin introducir las manos y los brazos, con ayuda de más personas podremos simular una interacción sin interrupciones entre usuarios, ya que la interacción de la persona con el dispositivo sin importar si es una mesa interactiva o una tableta, siempre será con los dedos.

La especificación técnica del dispositivo táctil más importante es:

Software:

- Sistema operativo: Android o MAC

3.3.5. Base de datos.

La base de datos cumple con el modelo relacional, que es uno de los más utilizados para sistemas para restaurantes que son los sistemas destinados a ser integrados con el tipo de aplicación a desarrollar, esto permite representar algunas entidades y procesos con el uso de tablas. Una de las ventajas de su implementación es el seguimiento de inventarios, procesar transacciones de comercio electrónico, administrar grandes cantidades de información del cliente, razón por la cual se seleccionó sobre las bases de datos no relacionales o mixtas.

El servicio de base de datos será ejecutado dentro del servidor de base de datos que es proporcionado por MARIADB.

En la figura 3.3. se muestra el diagrama entidad-relación propuesto para describir algunos de los elementos más importantes con los que debe de contar un sistema en su base de datos para que la implementación de la aplicación sea efectiva.

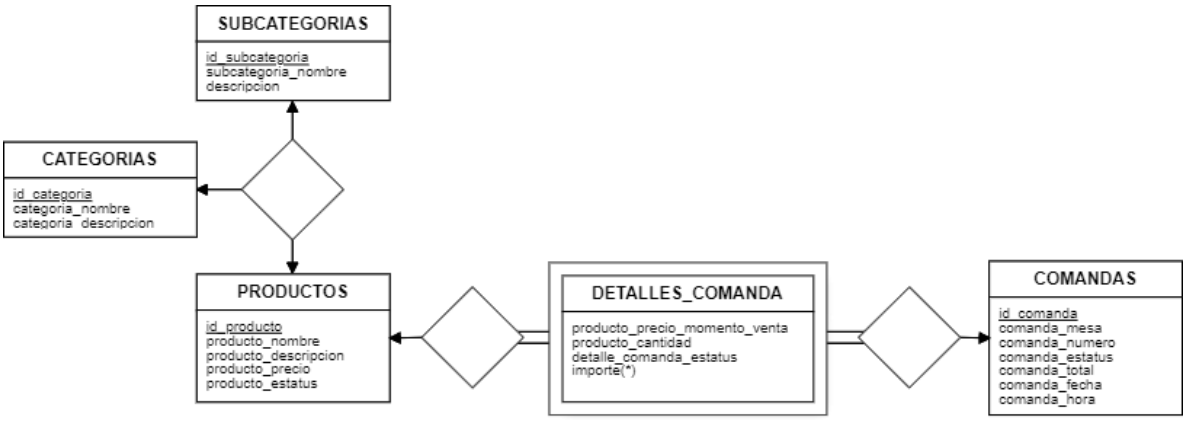


Figura 3.4. Diagrama Entidad-Relación. Cada entidad principal a involucrarse en el sistema se representa por una tabla.

Al diseñar la base de datos se incluyen cinco tablas, como mínimo será útil para el funcionamiento de la aplicación para registrar órdenes. A continuación, se describe el propósito de cada una:

Tabla categorías: Es la clasificación asignada a un producto a ofrecer por el restaurante. Por otro lado, el comensal podrá acceder a estas clasificaciones con la

aplicación para tener una perspectiva general de lo que podrá encontrar en el menú del restaurante. Los atributos definidos son los siguientes:

- id_categoria: Este atributo es un identificador único para cada registro de esta tabla.
- categoria_nombre: Define el nombre de la clasificación a primer nivel.
- categoría_descripcion: Describe la categoría de interés.

Tabla subcategoría: En caso de necesitar clasificar un producto con un nivel adicional, contamos con subcategorías, las cuales se encuentra dentro de las categorías. En esta tabla se puede encontrar los siguientes atributos:

- id_subcategoria: Este atributo es un identificador único para cada registro de esta tabla.
- subcategoria_nombre: Define el nombre de la clasificación a segundo nivel.
- subcategoria_descripcion: Describe la subcategoría de interés.

Tabla productos: Los alimentos y bebidas son considerados la oferta de un restaurante, los cuales son almacenados en esta tabla. Estos elementos se encuentran dentro de una categoría y pueden estar dentro de una subcategoría. Sabemos que un restaurante puede necesitar agregar otro nivel de clasificación, pero dichos niveles fueron limitados a subcategorías. Los atributos que son definidos en estas tablas son:

- id_producto: Este atributo es un identificador único para cada registro de esta tabla.
- producto_nombre: Define el nombre de cada elemento que ofrece el restaurante.
- producto_descripcion: Describe el producto de interés.
- producto_precio: Establece el costo monetario del elemento a ordenar.
- producto_estatus: Permite al restaurante habilitar o deshabilitar en el sistema un estatus en la base de datos para los elementos almacenados. En caso de que se cuente o no se cuente con dicho producto el restaurante podrá definir si aparece o no en el menú del comensal o bien en la interfaz de usuario.

Tabla comanda: Al hacer un pedido, los elementos solicitados por el comensal son registrados en una comanda. Esta comanda se encuentra disponible

para registrar todos los pedidos hechos durante la estancia en el restaurante. A veces es entregada como un recibo al cliente de su consumo total.

En esta tabla se registra los datos generales al final de la estancia del comensal en el restaurante, al realizar el pago algunos de estos datos son almacenados.

- id_comanda: Este atributo es un identificador único para cada registro de esta tabla.
- comanda_mesa: Define a que mesa tiene que ser entregado un pedido ya sea un alimento o bebida, aunque este elemento tiene que aparecer por pedido es colocado en la comanda, ya que todos los pedidos son requeridos por una mesa, entonces sería un dato muy repetitivo y de manera fácil se puede relacionar cada pedido o detalle con la comanda a la que pertenece en caso de requerirlo al hacer la entrega de los pedidos.
- comanda_numero: Cada mesa podrá desplegar cuatro menús o comanda, al momento de que se haga un pedido y en caso de hacer entrega de los productos podrán identificar qué comensal hizo el pedido en esa mesa. La aplicación ideal es con usuarios, pero eso no incluirá el desarrollo de la aplicación y se tendrá como una mejora a futuro.
- comanda_estatus: Define el estado de la comanda si se encuentra abierta o cerrada. Cuando se inicializa una comanda, tiene el estatus de abierto representado con el número 1, cuando el cliente termina su instancia y realiza el pago la comanda pasa a un estatus de cerrada representada con el número 0.
- comanda_total: Representa el monto total que el comensal tiene que pagar por lo consumido en su estancia en el restaurante.
- comanda_fecha: Representa la fecha en el cual fue registrado la comanda.
- comanda_hora: Representa la fecha en el cual fue registrado la comanda.

Tabla detalles comanda: Es conveniente registrar los detalles de cada pedido, es información útil para posteriores análisis por parte del restaurante. Los atributos involucrados en esta tabla son:

- producto_precio_momento_venta: Se registra el precio del producto, en el momento que fue hecho el pedido, ya que el precio puede ser actualizado para compras futuras.
- producto_cantidad: Define la cantidad del producto solicitado.
- detalle_comanda_estatus: Define el estatus del detalle de la comanda. Cuando se hace un pedido este detalle tiene el estatus de abierto representado con el número 1, esto permite principalmente a los empleados saber qué pedidos tienen que ser atendidos, ya que son atendidos el detalle pasa a un estatus de cerrado representado con el número 0.
- importe: Es el valor económico calculado del precio en el momento de la venta y la cantidad del producto, y este importe es calculado por detalle. Para al final cuando el comensal termine su estancia en el restaurante y proceda a pagar, estos importes serán sumados, así obteniendo el monto total de la comanda.

3.3.6. API REST.

La API será una interfaz a desarrollar para comunicar las interfaces de usuarios de la aplicación con la base de datos. La tecnología a utilizar es el lenguaje de programación PYTHON, así como su microframework FLASK. FLASK se encarga de inicializar un servidor al momento en el que se ejecuta un script de PYTHON.

La figura 3.3. muestra el diagrama de clases que representa el diseño de la API REST. Este servicio será ejecutado dentro del servidor web, razón por la cual se está utilizando este tipo de tecnología.

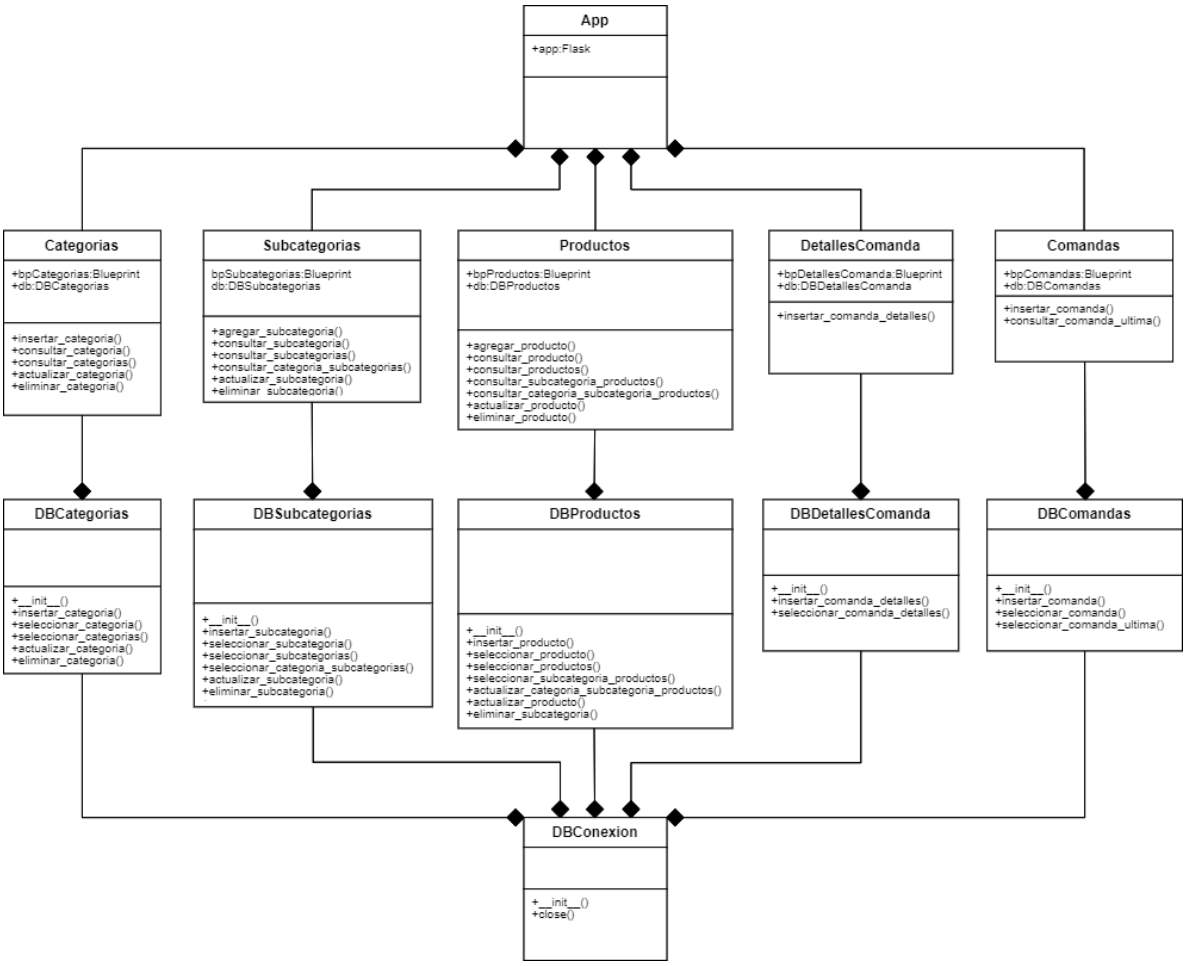


Figura 3.5. Diagrama de clases. Muestra la interacción entre las clases a utilizar en el desarrollo de la aplicación.

A continuación, se describe el uso de cada una de las clases:

App: Es la clase principal, donde se instancia un objeto de nombre app tipo Flask, este objeto permite iniciar el servidor web. Además con el método “register_blueprint()” asocia las rutas que podrán ser consultadas por HTTP desde las clases “Categorías”, “Subcategorías”, “Productos”, “DetallesComanda”, “Comandas”.

DBConexion: Es la clase que permite generar instancias de la conexión a la base de datos y poder generar operaciones SQL.

DBCategorias: Esta clase genera una instancia de nombre “con” de la clase “DBConexion” que es la conexión entre la API y la base de datos, al utilizar los

métodos de esta clase podrá realizar operaciones sobre la base de datos. Esta clase es instanciada desde la clase “Categorías”.

DBSubcategorias: Esta clase genera una instancia de nombre “con” de la clase “DBConexion” que es la conexión entre la API y la base de datos, al utilizar los métodos de esta clase podrá realizar operaciones sobre la base de datos. Esta clase es instanciada desde la clase “Subcategorias”.

DBProductos: Esta clase genera una instancia de nombre “con” de la clase “DBConexion” que es la conexión entre la API y la base de datos, al utilizar los métodos de esta clase podrá realizar operaciones sobre la base de datos. Esta clase es instanciada desde la clase “Productos”.

DBDetallesComanda: Esta clase genera una instancia de nombre “con” de la clase “DBConexion” que es la conexión entre la API y la base de datos, al utilizar los métodos de esta clase podrá realizar operaciones sobre la base de datos. Esta clase es instanciada desde la clase “DetallesComanda”.

DBComandas: Esta clase genera una instancia de nombre “con” de la clase “DBConexion” que es la conexión entre la API y la base de datos, al utilizar los métodos de esta clase podrá realizar operaciones sobre la base de datos. Esta clase es instanciada desde la clase “Comandas”.

Categorias: En esta clase instancia un objeto de nombre “bpCategorias” que permite el registro de la clase dentro de un “Blueprint” de FLASK para que esta clase sea administrada por este componente. Además de generar las rutas que podrán ser solicitadas desde un cliente web con el uso del protocolo HTTP. Estas rutas están asociadas a métodos que no son visibles para el cliente, estos métodos hacen uso de un objeto instanciado de nombre “db” que fueron definidos desde la clase “DBCategorias” que nos retornan los resultados de las consultas SQL. Los métodos de la clase “Categorias” retornan mensajes en formato json para contestar al cliente web.

Subcategorias: En esta clase instancia un objeto de nombre “bpSubcategorias” que permite el registro de la clase dentro de un “Blueprint” de FLASK para que esta clase sea administrada por este componente. Además de generar las rutas que podrán ser solicitadas desde un cliente web con el uso del

protocolo HTTP. Estas rutas están asociadas a métodos que no son visibles para el cliente, estos métodos hacen uso de un objeto instanciado de nombre “db” que fueron definidos desde la clase “DBSubcategorias” que nos retornan los resultados de las consultas SQL. Los métodos de la clase “Subcategorias” retornan mensajes en formato json para contestar al cliente web.

Productos: En esta clase instancia un objeto de nombre “bpProductos” que permite el registro de la clase dentro de un “Blueprint” de FLASK para que esta clase sea administrada por este componente. Además de generar las rutas que podrán ser solicitadas desde un cliente web con el uso del protocolo HTTP. Estas rutas están asociadas a métodos que no son visibles para el cliente, estos métodos hacen uso de un objeto instanciado de nombre “db” que fueron definidos desde la clase “DBProductos” que nos retornan los resultados de las consultas SQL. Los métodos de la clase “Productos” retornan mensajes en formato json para contestar al cliente web.

DetallesComanda: En esta clase instancia un objeto de nombre “bpDetallesComanda” que permite el registro de la clase dentro de un “Blueprint” de FLASK para que esta clase sea administrada por este componente. Además de generar las rutas que podrán ser solicitadas desde un cliente web con el uso del protocolo HTTP. Estas rutas están asociadas a métodos que no son visibles para el cliente, estos métodos hacen uso de un objeto instanciado de nombre “db” que fueron definidos desde la clase “DBDetallesComanda” que nos retornan los resultados de las consultas SQL. Los métodos de la clase “DetallesComanda” retornan mensajes en formato json para contestar al cliente web.

Comandas: En esta clase instancia un objeto de nombre “bpComandas” que permite el registro de la clase dentro de un “Blueprint” de FLASK para que esta clase sea administrada por este componente. Además de generar las rutas que podrán ser solicitadas desde un cliente web con el uso del protocolo HTTP. Estas rutas están asociadas a métodos que no son visibles para el cliente, estos métodos hacen uso de un objeto instanciado de nombre “db” que fueron definidos desde la clase “DBComandas” que nos retornan los resultados de las consultas SQL. Los métodos

de la clase “Comandas” retornan mensajes en formato json para contestar al cliente web.

3.3.7. Aplicación multitáctil multiusuario.

La aplicación multitáctil multiusuario se tiene que implementar dentro del dispositivo táctil. Esta aplicación al ser ejecutada será un cliente web que se mantendrá en comunicación con el servidor local o en la nube con el uso de drives de red del dispositivo, además de mantenerse comunicado con los servicios destinados para el funcionamiento de la aplicación como lo es el servidor de bases de datos y el servidor web. En su conjunto se podrá realizar el registro de órdenes en restaurantes.

La aplicación se tiene que diseñar pensando en el uso que se le va a dar junto al dispositivo y se debe de pensar en la cantidad de personas que estarán interactuando simultáneamente.

La implementación de esta tecnología multitáctil multiusuario será en una mesa, precisamente una mesa interactiva y se va a limitar el uso de 4 usuarios simultáneos. Se tiene que diseñar una interfaz de usuario utilizando ventanas emergentes que es lo recomendado para tecnologías multitáctil multiusuario. La figura 3.4. muestra el diseño de la UI (Interfaz de usuario) sin usuarios interactuando, la figura 3.5. muestra el diseño de la UI con usuarios interactuando y la figura 3.6. muestra el diseño de las ventanas emergentes.

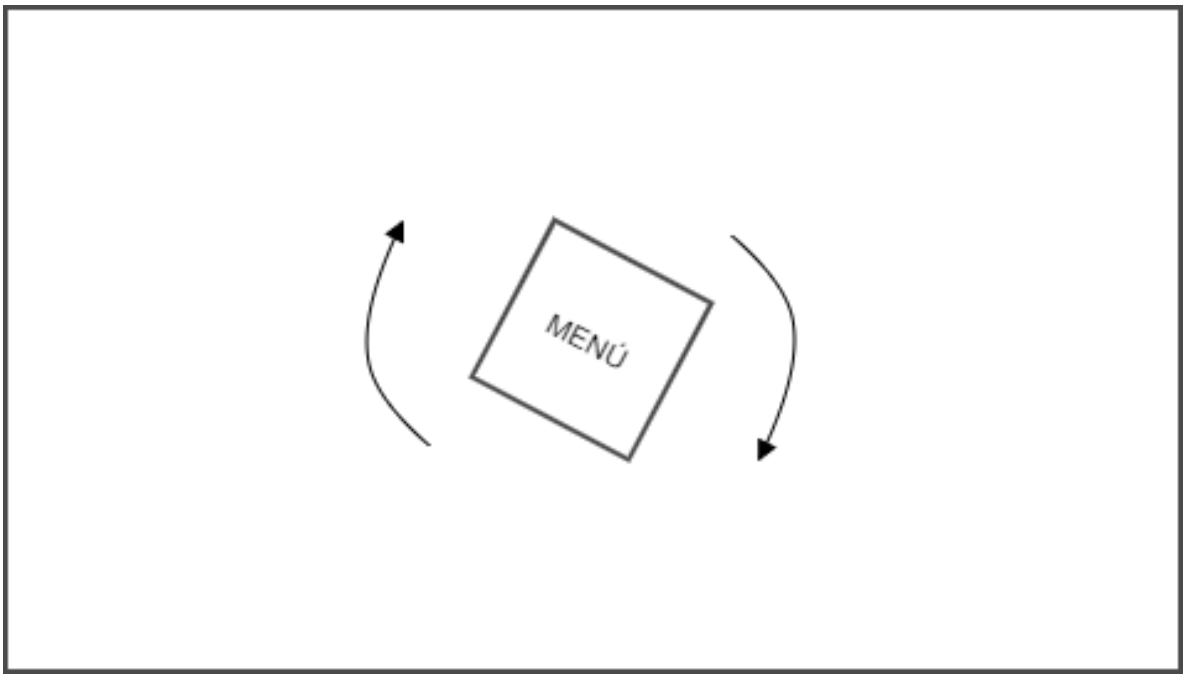


Figura 3.6. UI de aplicación multitáctil multiusuario sin usuarios interactuando.

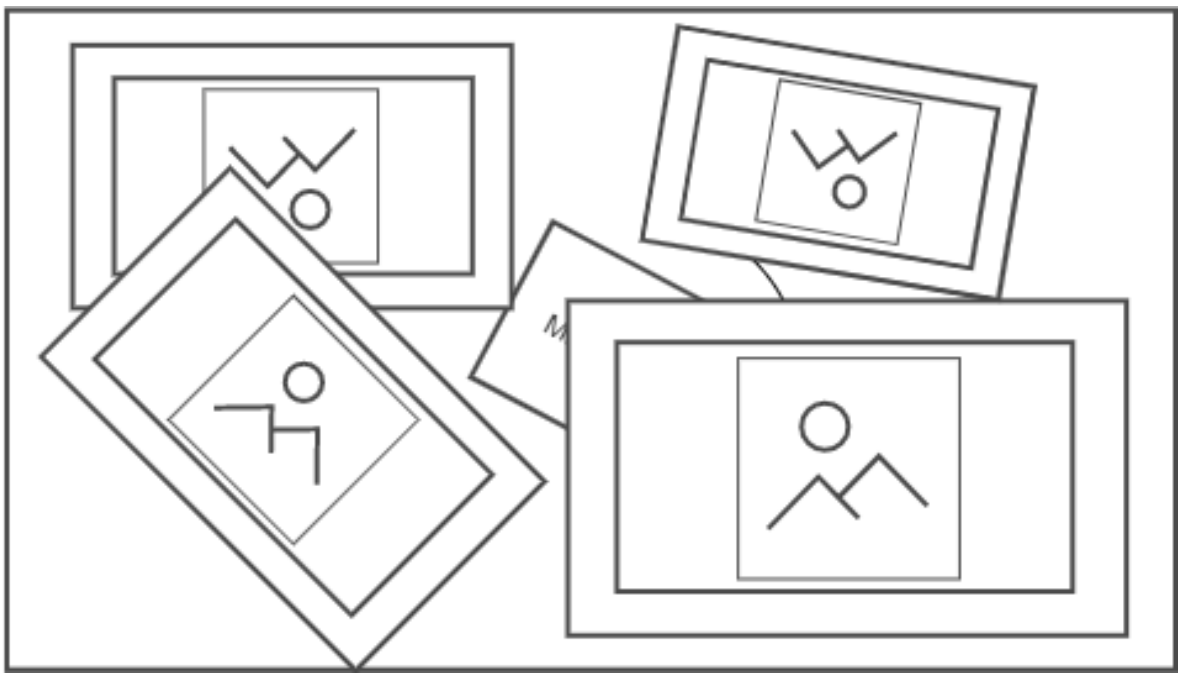


Figura 3.7. UI de aplicación multitáctil multiusuario con usuarios interactuando.

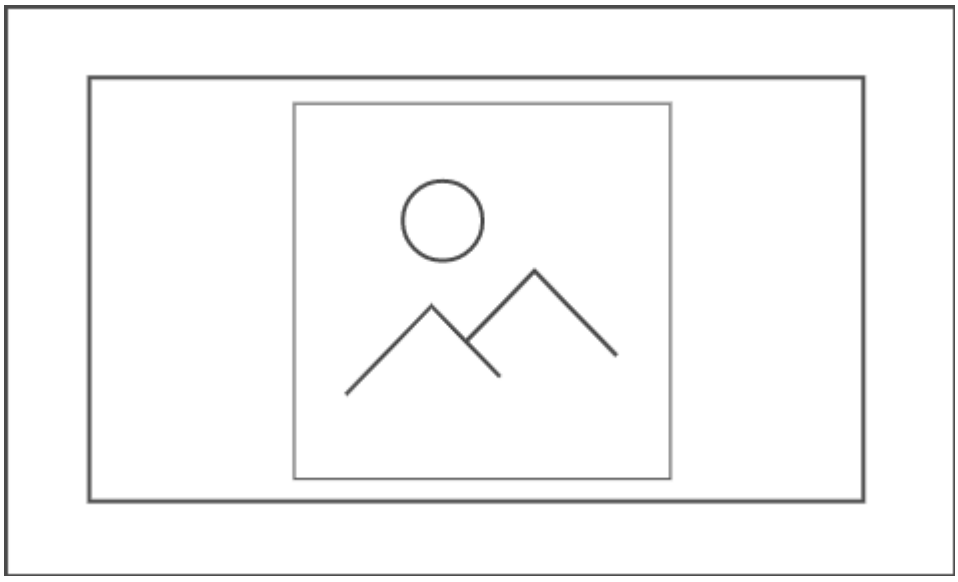


Figura 3.8. Diseño de las ventanas emergentes.

3.4. Arquitectura del sistema.

Se propone implementar una arquitectura con tres componentes principales. Una base de datos y un servicio web ambos con sus respectivos servicios, podrán ser alojados en un servidor local o en la nube. Además de una aplicación multitáctil multiusuario que podrá ser instalado en un dispositivo táctil, es ideal que el dispositivo sea una mesa interactiva para su implementación con el usuario final. En caso de prueba se puede utilizar cualquier dispositivo móvil, en este caso se recomienda el uso de una tableta.

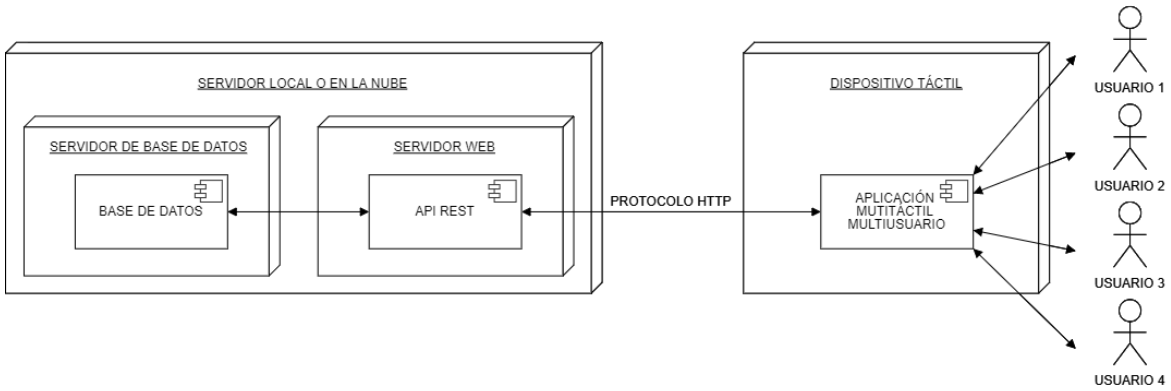


Figura 3.9. Arquitectura del sistema. Muestra la organización de los elementos principales, así mismo la interacción entre dichos elementos. La aplicación a desarrollar y los servicios con los que va a interactuar como lo es el servidor web y el servidor de base de datos se comunicarán utilizando el protocolo HTTP.

3.4.1. Módulos de aplicación multitáctil multiusuario.

La figura 3.9. describe los módulos principales, sus submódulos y como interactúan entre sí. Un módulo se utilizará como una división lógica de la aplicación y los submódulos serán elementos que hace funcionar la aplicación con el uso de código escrito en C#.

- **Los módulos principales son:**
- **UI (Interfaz de usuario):** El módulo UI contendrá los submódulos que se encargarán de la funcionalidad de las interfaces de usuario. Estos son elementos visuales que se instanciarán por referencia por el módulo de servicios y por el manage UI.
- **Manage UI:** Los elementos que encontramos en manage UI, administrarán los elementos visuales, según los eventos ejecutados por estos. Desde dibujar en pantalla los que el usuario necesita ver, hasta administrar los eventos ejecutados por los elementos visuales.
- **Modelos:** Es la estructura de los objetos, la cual podrá ser de ayuda para mantener en memoria la información recuperada de las APIs y la información mostrada a través de la UI.
- **Servicios:** Cuando los eventos visuales serán llamados por el manage UI, a su vez serán llamados por las APIs que se encargan de recuperar la

información del servidor. La información recuperada es mostrada atreves de los ScrollView.

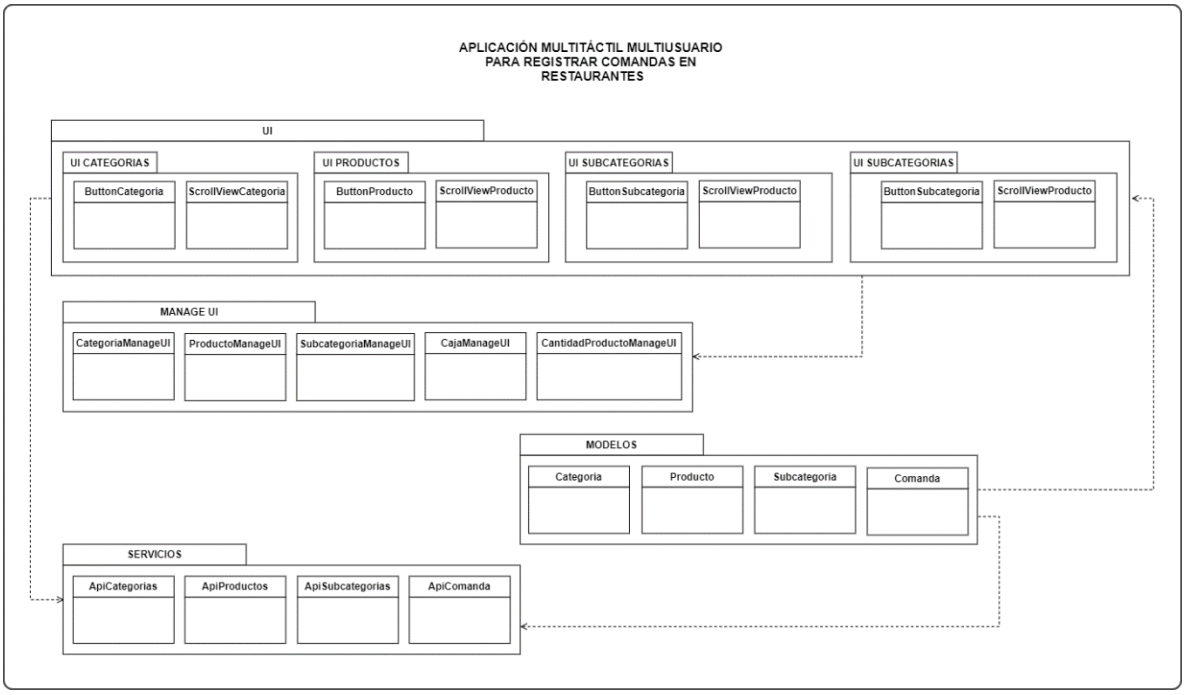


Figura 3.10. Módulos de aplicación multitáctil multiusuario. Estos módulos interactúan entre si pasando la referencia de la información de los submódulos que contienen. Cada ventana emergente contendrá todos los módulos mostrados en la figura, donde se creará una instancia en memoria independiente a otros generados.

Capítulo 4.

4. Implementación de la aplicación.

4.1. Base de datos.

La base de datos será implementada con el uso de QUERIES, serán utilizados para definir la estructura de la base de datos. Ya que la manipulación de la información será realizada con el uso de un cliente web.

El QUERY de la figura 4.1. crea la base de datos, en este caso se ha nombrado “project”. Además, se pone en uso para realizar operaciones sobre la base de datos.

```
CREATE DATABASE project;

USE project;
```

Figura 4.1. Crear y usar base de datos.

La figura 4.2. Muestra cómo crear la tabla de “categorías” y la definición de su estructura.

```
CREATE TABLE categorias(
    id_categoria INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    categoria_nombre VARCHAR(50),
    categoria_descripcion VARCHAR(150)
);
```

Figura 4.2. Crear tabla categorías.

La figura 4.3. Muestra cómo crear la tabla de subcategorías y la definición de su estructura.

```
CREATE TABLE subcategorias(
    id_subcategoria INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    subcategoria_nombre VARCHAR(50),
    subcategoria_descripcion VARCHAR(150),
);
```

Figura 4.3. Crear tabla subcategorías.

La figura 4.4. Muestra cómo crear la tabla de “productos” y la definición de su estructura. En la tabla de productos encontramos dos identificadores externos como llaves foráneas, el id de las categorías y el id de las subcategorías, esto se debe, a que en la definición del modelo entidad-relación de la base de datos se diseñó una relación ternaria entre las tablas de categoría, subcategorías y productos y se tendría que generar una tabla con los identificadores de las 3 tablas, pero en vez de eso se optó por usar la tabla de productos para almacenar los identificadores de la relación ternaria.

```
CREATE TABLE productos(
    id_producto INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    producto_nombre VARCHAR(50),
    producto_descripcion VARCHAR(100),
    producto_precio FLOAT,
    producto_estatus INT,
    id_categoria INT,
    id_subcategoria INT,
    FOREIGN KEY (id_categoria) REFERENCES categorias(id_categoria),
    FOREIGN KEY (id_subcategoria) REFERENCES subcategorias(id_subcategoria)
);
```

Figura 4.4. Crear tabla productos.

La figura 4.5. Muestra cómo crear la tabla de “comandas” y la definición de su estructura.

```
CREATE TABLE comandas(
    id_comanda INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    comanda_mesa INT,
    comanda_numero INT,
    comanda_estatus INT,
    comanda_total FLOAT,
    comanda_fecha DATE,
    comanda_hora TIME
);
```

Figura 4.5. Crear tabla comandas.

La figura 4.6. Muestra cómo crear la tabla de “detalles_comanda” y la definición de su estructura.

```
CREATE TABLE detalles_comanda(
    id_producto INT,
    id_comanda INT,
    producto_precio_momento_venta FLOAT,
    producto_cantidad INT,
    importe FLOAT,
    PRIMARY KEY(id_producto,id_comanda)
);
```

Figura 4. 6. Crear tabla detalles_comanda.

En la figura 4.7 muestra el uso de XAMPP para inicializar el servicio de base de datos y utilizar PHPMYADMIN para acceder a la base de datos. Se puede utilizar la base de datos sin el servicio de APACHE y acceder directo desde la terminal, en este proyecto se implementó de esta forma por la rapidez de trabajo.

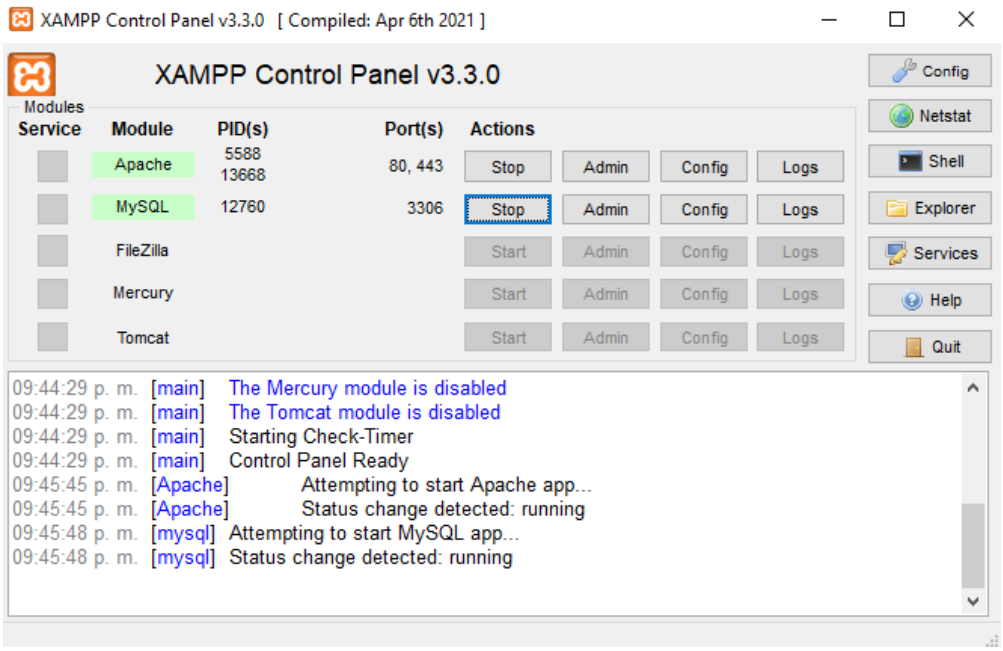


Figura 4.7. Iniciar el servidor.

La figura 4.8. muestra los datos del servidor, estos son proporcionados por la interfaz de PHPMYADMIN y podemos encontrar que el servidor de bases de datos se encuentra en modo local y el sistema gestor de base de datos es MARIADB.

Figura 4.8. Datos del servidor.

Al desplegar el contenido de la base de datos, se puede confirmar que la implementación fue realizada con éxito.

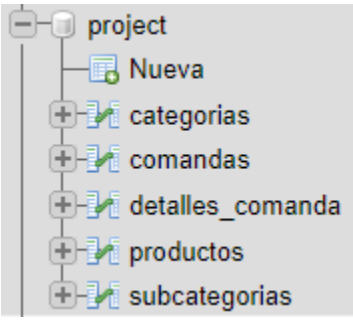


Figura 4.9. Tablas generadas a partir de los querys.

4.2. API REST.

La API REST fue implementada con PYTHON y su microframework FLASK. En la figura 4.10. Podemos ver la distribución del proyecto. Debido al que el proyecto no se enfoca al desarrollo del sistema de registro de órdenes en restaurante y sí en el desarrollo de la aplicación. El código está listo para escalar en caso de utilizar la API REST desarrollada en este proyecto.

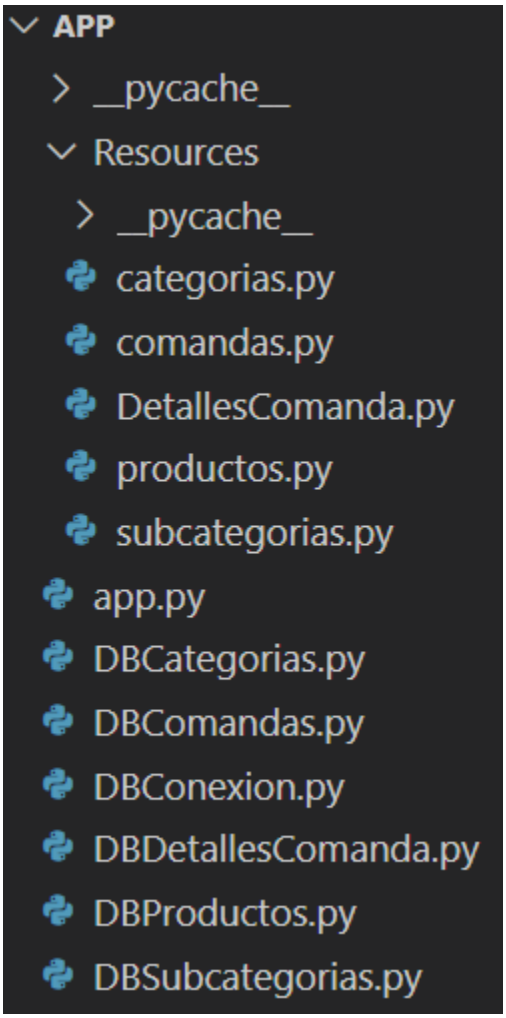


Figura 4.10. Archivos para desarrollar API en PYTHON con FLASK.

La figura 4.11 muestra el archivo App.py que a pesar de no contener una clase definida, es el archivo que se utiliza como el proceso principal, ya que FLASK inicializa el proyecto en este archivo con el método “run()”. Podemos ver como al objeto app registra atributos importados de otros archivos, esto es para mantener el código limpio y sea fácilmente escalable sin la necesidad de hacer instancias de las clases que se encuentran dentro de esos archivos. De eso ya se encarga el módulo blueprint de FLASK.

```python
from flask import Flask
from Resources.categorias import bpCategorias
from Resources.subcategorias import bpSubcategorias
from Resources.productos import bpProductos
from Resources.comandas import bpComandas
from Resources.DetallesComanda import bpDetallesComanda

app = Flask(__name__)
app.register_blueprint(bpCategorias)
app.register_blueprint(bpSubcategorias)
app.register_blueprint(bpProductos)
app.register_blueprint(bpComandas)
app.register_blueprint(bpDetallesComanda)

if __name__ == '__main__':
    app.run(debug=True)
```

Figura 4.11. Clase App.

La figura 4.12. Muestra la clase DBConexion, esta clase administra la conexión con la base de datos, y pone a distribución mecanismos para realizar transacciones sobre la base de datos para las clases que lo requieran. Esta clase está hecha en un archivo aparte en donde se realizan las consultas sql para mantener código limpio y escalable.

```python
import pymysql

class DBConexion:
    def __init__(self):
        self.connection = pymysql.connect(
            host = 'localhost',
            user = 'root',
            password = '',
            db = 'project'
        )
        self.cursor = self.connection.cursor()
        print('Conexion establecida exitosamente')

    def close(self):
        self.connection.close()
```

Figura 4.12. Clase “DBConexion”.

La figura. 4.13. es la clase “DBCategorias” que se encarga de realizar transacciones en la base de datos, aquí solo se involucra a la entidad “categoria”, puede ser uno o una lista de este elemento.

```python
from DBConexion import DBConexion

class DBCategorias:
    def __init__(self):
        self.con = DBConexion()

    def insertar_categoria(self,categoriaNombre,categoriaDescripcion):...

    def seleccionar_categoria(self,idCategoria):...

    def seleccionar_categorias(self):...

    def actualizar_categoria(self,idCategoria, categoriaNombre,categoriaDescripcion):

    def eliminar_categoria(self,idCategoria):...
```

Figura 4.13. Clase “DBCategorias”.

La figura. 4.14. es la clase “DBSubcategorias” que se encarga de realizar transacciones en la base de datos, aquí sólo se involucra a la entidad “subcategoria”, puede ser uno o una lista de este elemento.

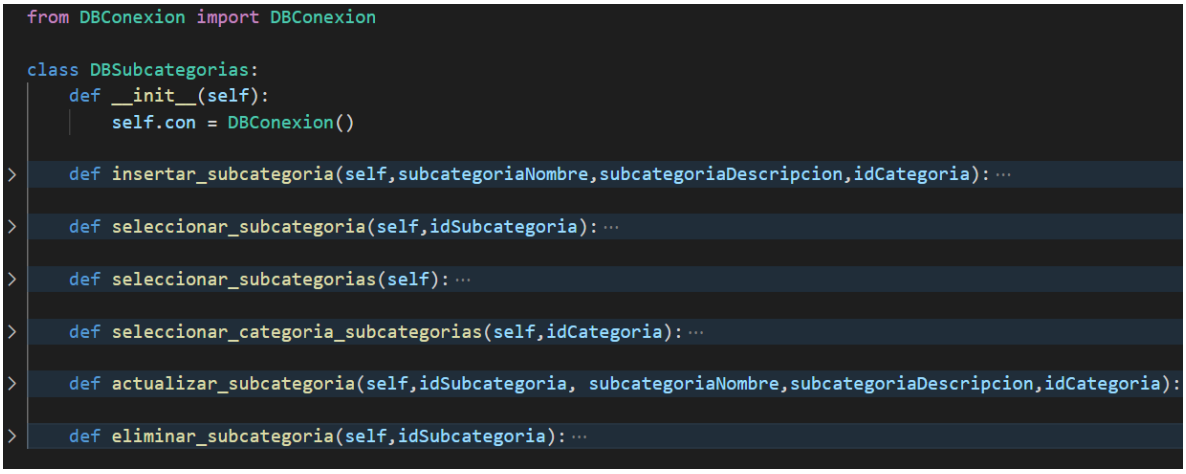


Figura 4.14. Clase “DBSubcategorias”.

La figura. 4.15. es la clase “DBProductos” que se encarga de realizar transacciones en la base de datos, aquí sólo se involucra a la entidad “producto”, puede ser uno o una lista de este elemento.

Figura 4.15. Clase “DBProductos”.

La figura. 4.16. es la clase “DBDetallesComanda” que se encarga de realizar transacciones en la base de datos, aquí sólo se involucra a la entidad “detalle_comanda”, puede ser uno o una lista de este elemento.

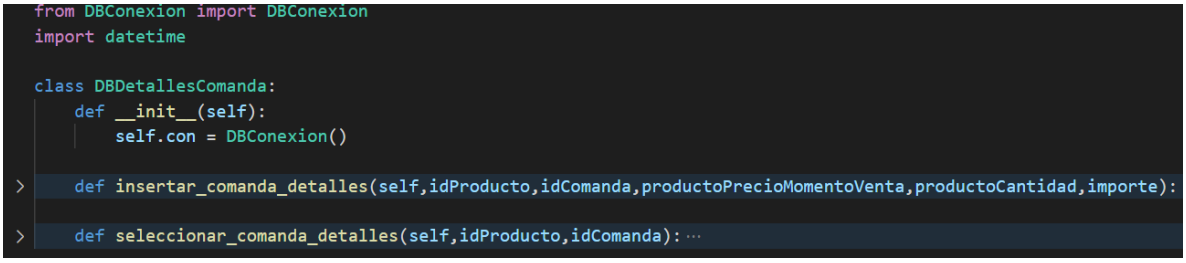


Figura 4.16. Clase “DBDetallesComanda”.

La figura. 4.17 es la clase “DBComandas” que se encarga de realizar transacciones en la base de datos, aquí sólo se involucra a la entidad “comanda”, puede ser uno o una lista de este elemento.

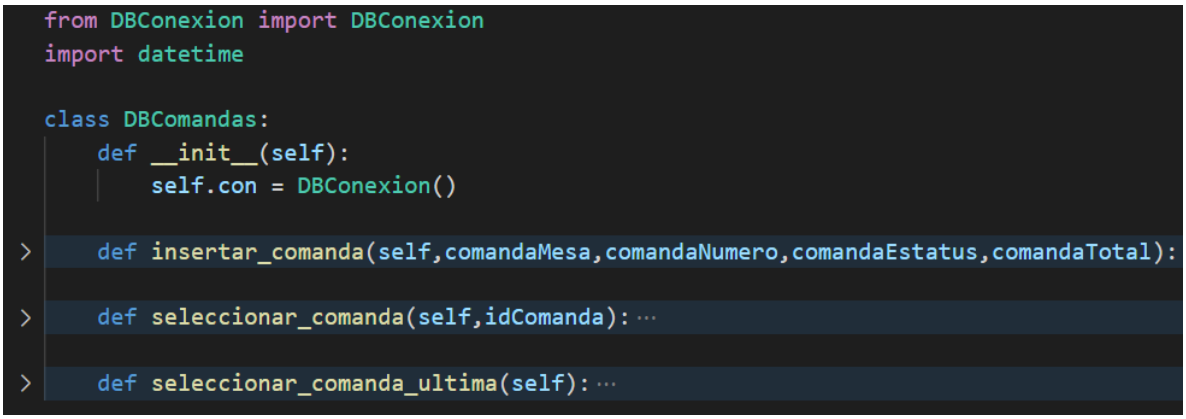


Figura 4.17. Clase “DBComandas”.

La figura. 4.18. es la clase “Categorias”, en este archivo no se define una clase propiamente, ya que está administrado por el módulo blueprint. Esta clase tiene como finalidad proporcionar a los clientes web que consuman este servicio las rutas necesarias en este caso para operar sobre la entidad “categorías”. Ofrece métodos “POST”, “GET”, “PUT” y “DELETE” del protocolo HTTP.

```python
from flask import request, jsonify, Blueprint
from DBCategorias import DBCategorias

bpCategorias = Blueprint('routes-categorias',__name__)
db = DBCategorias()

@bpCategorias.route('/insertar/categoria',methods=['POST'])
def insertar_categoria():···

@bpCategorias.route('/consultar/categoria',methods=['GET'])
def  consultar_categoria():···

@bpCategorias.route('/consultar/categorias',methods=['GET'])
def  consultar_categorias():···

@bpCategorias.route('/actualizar/categoria',methods=['PUT'])
def actualizar_categoria():···

@bpCategorias.route('/eliminar/categoria',methods=['DELETE'])
def eliminar_categoria():···
```

Figura 4.18. Clase “Categoria”.

La figura. 4.19. es la clase “Subcategorias”, en este archivo no se define una clase propiamente ya que es administrado por el módulo blueprint. Esta clase tiene como finalidad proporcionar a los clientes web que consuman este servicio las rutas necesarias en este caso para operar sobre la entidad “subcategorias”. Ofrece métodos “POST”, “GET”, “PUT” y “DELETE” del protocolo HTTP.

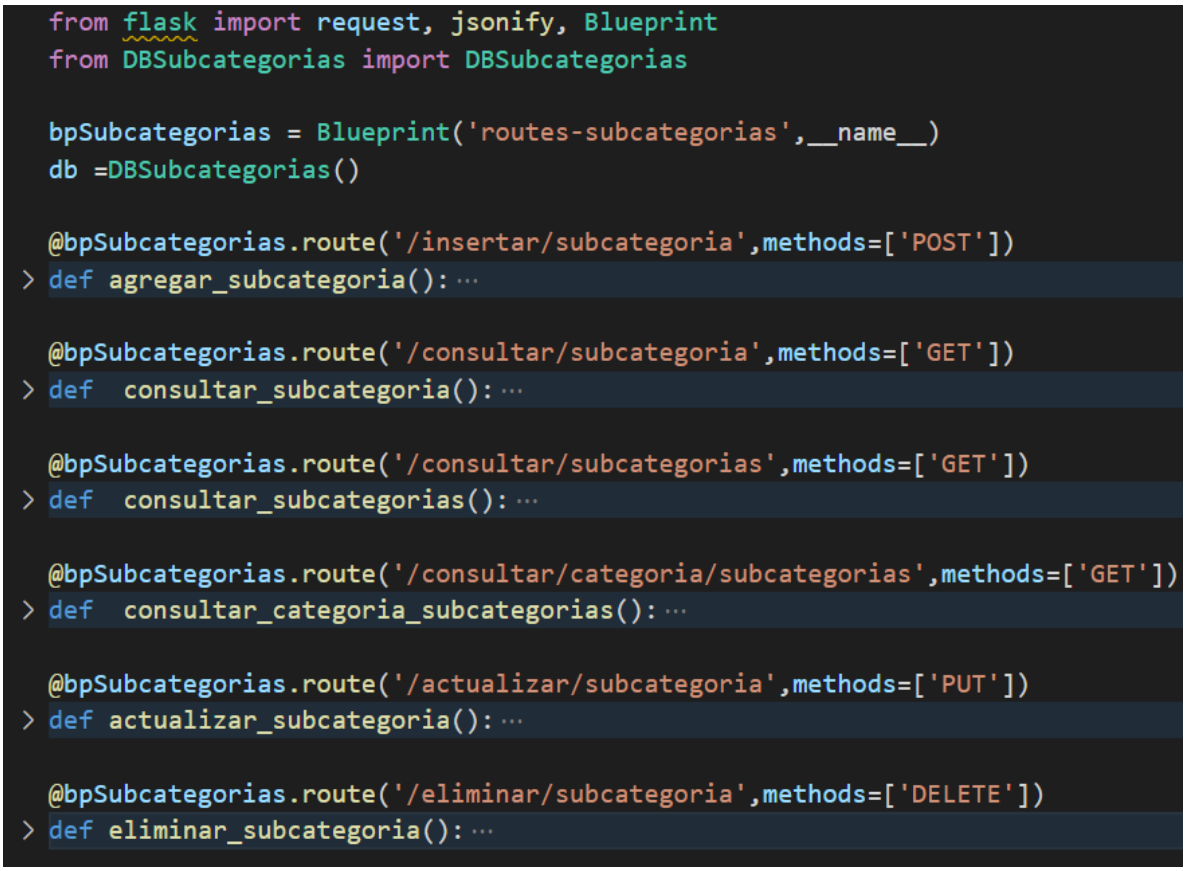


Figura 4.19. Clase “Subcategoria”.

La figura. 4.20. es la clase “Productos”, en este archivo no se define una clase propiamente, ya que está administrado por el módulo blueprint. Esta clase tiene como finalidad proporcionar a los clientes web que consuman este servicio las rutas necesarias en este caso para operar sobre la entidad “productos”. Ofrece métodos “POST”, “GET”, “PUT” y “DELETE” del protocolo HTTP.

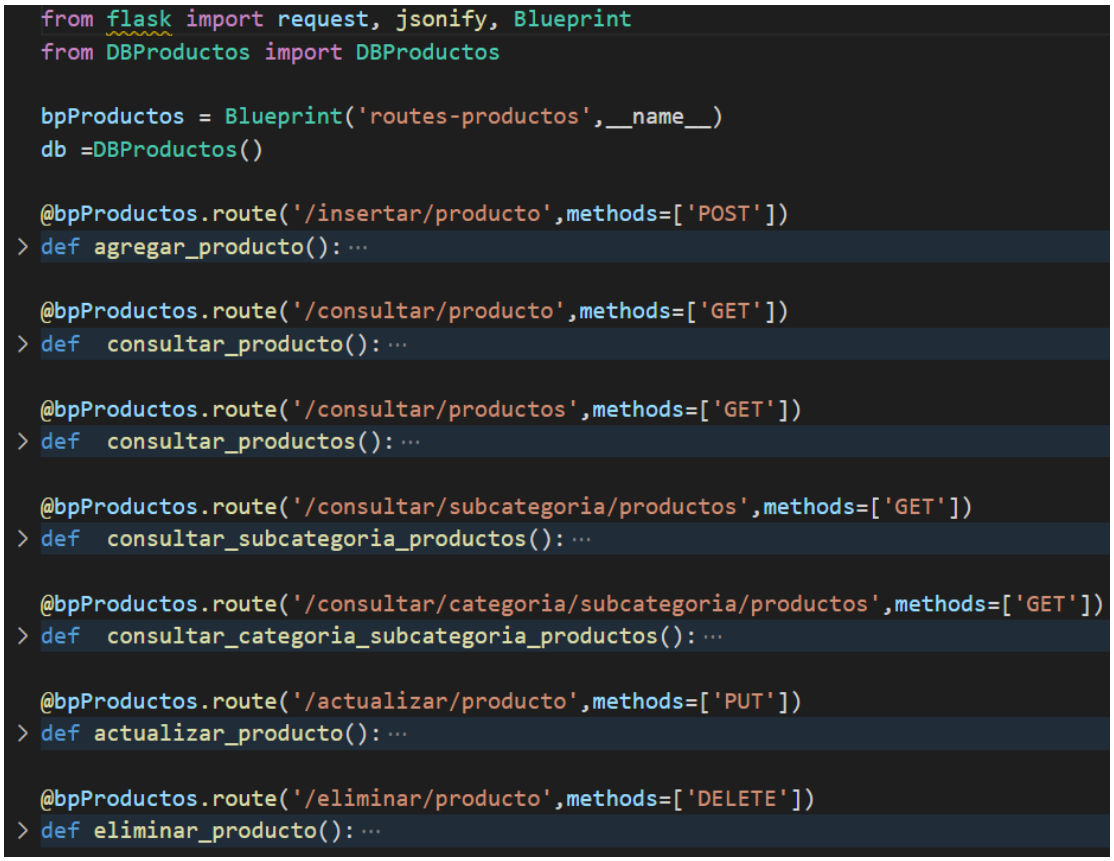


Figura 4.20. Clase “Productos”.

La figura. 4.21. es la clase “DetallesComanda”, en este archivo no se define una clase propiamente, ya que está administrado por el módulo blueprint. Esta clase tiene como finalidad proporcionar a los clientes web que consuman este servicio las rutas necesarias en este caso para operar sobre la entidad “detalles_comanda”. Ofrece el método “POST” del protocolo HTTP.

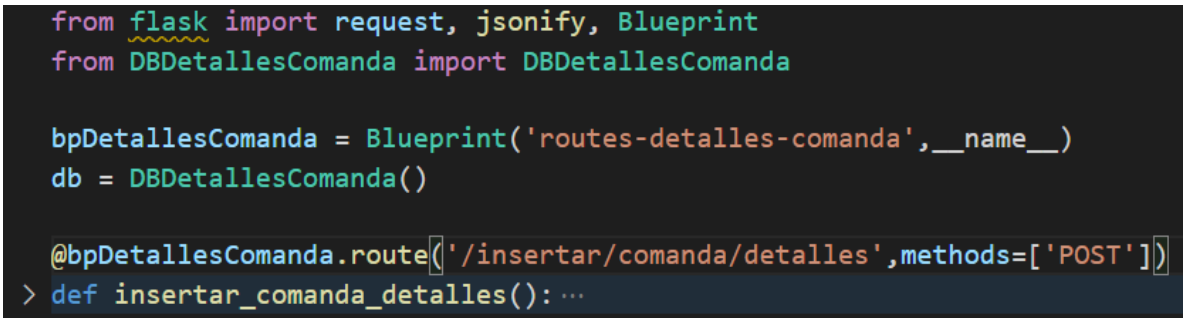


Figura 4.21. Clase “DetallesComanda”.

La figura. 4.22. es la clase “Comandas”, en este archivo no se define una clase propiamente, ya que está administrado por el módulo blueprint. Esta clase tiene como finalidad proporcionar a los clientes web que consuman este servicio las rutas necesarias en este caso para operar sobre la entidad “comandas”. Ofrece métodos “POST”, “GET” del protocolo HTTP.

```
from flask import request, jsonify, Blueprint
from DBComandas import DBComandas

bpComandas = Blueprint('routes-comandas',__name__)
db = DBComandas()

@bpComandas.route('/insertar/comanda',methods=['POST'])
def insertar_comanda():···

@bpComandas.route('/consultar/comanda/ultima',methods=['GET'])
def  consultar_comanda_ultima():···
```

Figura 4.22. Clase “Comandas”.

En la figura 4.23 y 4.24 se puede observar cómo al inicializar los servicios en PYTHON el servidor inicia sin ningún problema.

```
(project) C:\Users\HP-PAVILION\Desktop\python\project\App>python app.py
```

Figura 4.23. Iniciar servicio en Python.

```
* Debugger is active!
* Debugger PIN: 204-447-038
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figura 4.24. Servidor en funcionamiento.

POSTMAN es un programa que hace la función de un cliente web, es utilizado en el proyecto para remplazar las operaciones que haría un software administrativo que permita la administración de la oferta del restaurante, además es de gran ayuda para validar las APIS desarrolladas. Podemos ver la interfaz del programa utilizado en la figura 4.25.

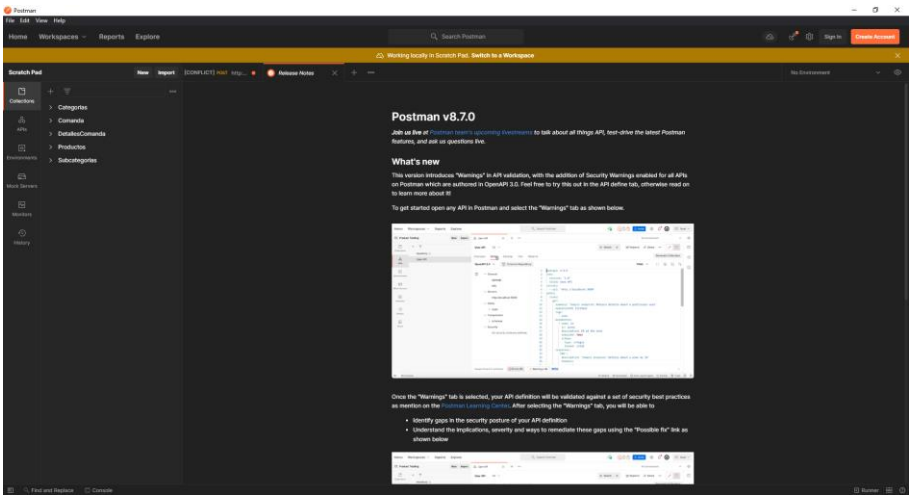


Figura 4.25. POSTMAN.

En la figura 4.26. se muestra algunas peticiones web ya preparadas en POSTMAN, como lo son “GET”, “POST”, “PUT” y “DELETE”.

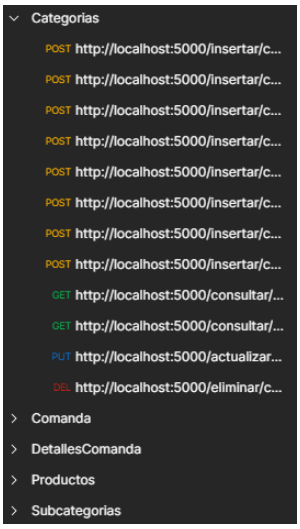


Figura 4.26. Peticiones HTTP por POSTMAN.

Para comprobar el uso de la API, se va a utilizar POSTMAN para hacer peticiones HTTP al servidor. Se realizarán pruebas con entidad “categorías” por lo que se hace una consulta a la tabla para verificar su contenido y compararlo con los resultados que arroje la API. En la figura 4.27. se puede observar el contenido de la tabla a trabajar.

Mostrando filas 0 - 7 (total de 8, La consulta tardó 0,0020 segundos.)

```
SELECT * FROM `categorias`;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 | Filtrar filas: Buscar en esta tabla | Sort by key: Ninguna

+ Opciones

	id_categoria	categoria_nombre	categoria_descripcion
Editar Copiar Borrar	1	SIN CATEGORÍA	SIN CATEGORÍA
Editar Copiar Borrar	2	ENTRADAS	PARA INICIAR
Editar Copiar Borrar	3	PLATILLOS	PLATO FUERTE
Editar Copiar Borrar	4	MENÚ KIDS	PARA LOS PEQUEÑOS
Editar Copiar Borrar	5	POSTRES	PARA FINALIZAR
Editar Copiar Borrar	6	ESPECIALIDADES	PARA EL ANTOJO
Editar Copiar Borrar	7	BEBIDAS	PARA ACOMPAÑAR
Editar Copiar Borrar	8	CATEGORÍA DE PRUEBA PARA ELIMINAR	CATEGORÍA DE PRUEBA PARA ELIMINAR

Figura 4.27. Contenido de la tabla “categorías” en la base de datos.

En la figura 4.28. se observa el resultado que arroja POSTMAN al realizar una petición GET con un parámetro a la tabla de “categorías”, retorna un objeto

JSON. Además, nos arroja un estatus “200 OK”, que eso significa que la operación se realizó con éxito.

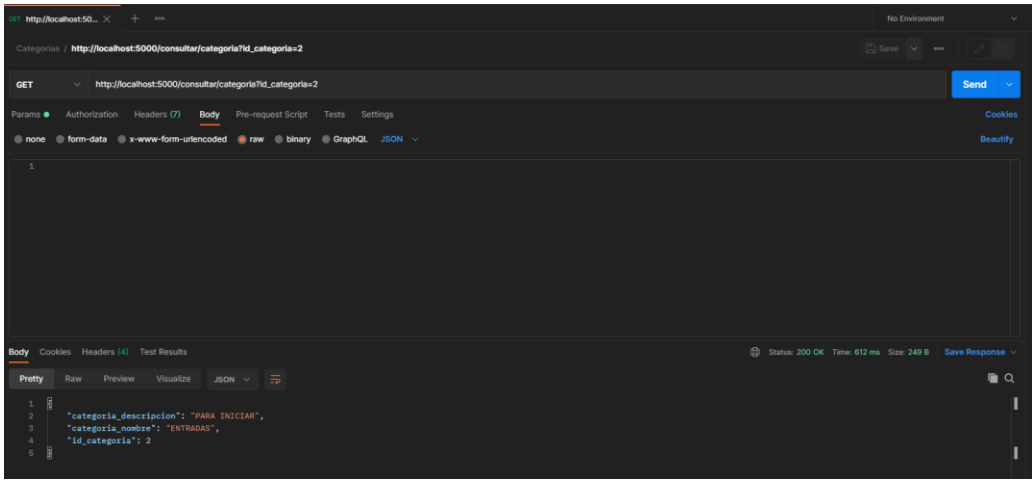


Figura 4.28. GET en POSTMAN para obtener una categoría.

En la figura 4.29. y 4.30. POSTMAN realiza de nuevo una petición GET sin parámetros, la dirección de la API a solicitar esta vez es distinta a la anterior, ya que requiere obtener una lista de objetos JSON con todas las categorías existentes.

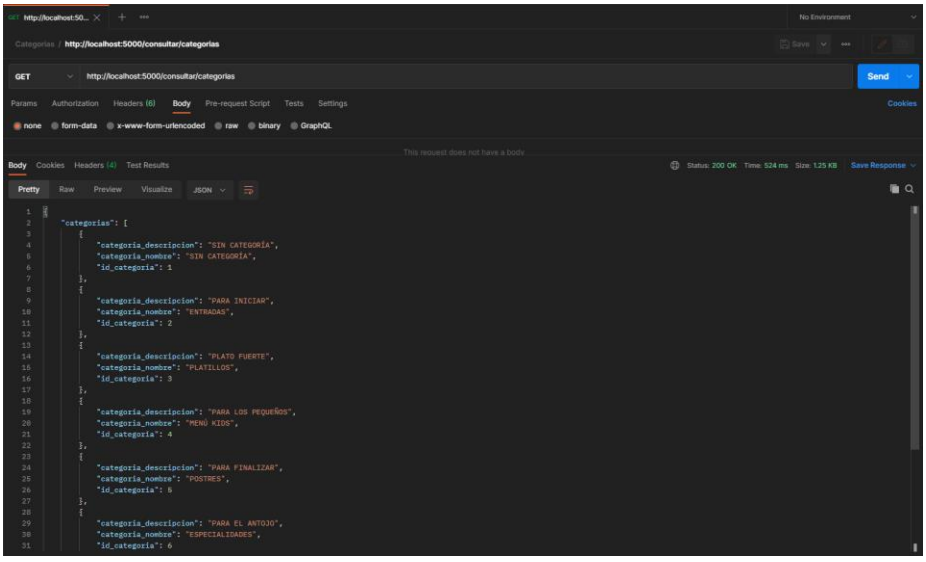


Figura 4.29. GET en POSTMAN para obtener una lista de categorías.

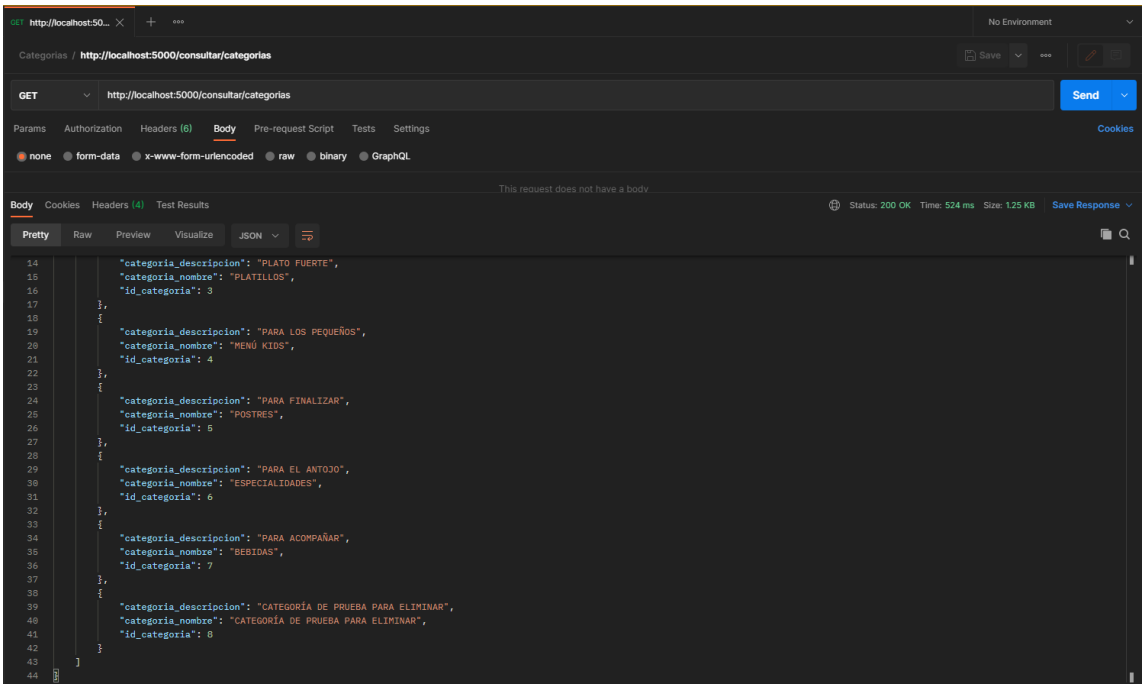


Figura 4.30. GET en POSTMAN para obtener una lista de categorías.

Todas estas peticiones tienen que ser procesadas por el servidor, por lo cual el servidor responde a dichas peticiones. Esto se puede observar al monitorear la terminal donde tenemos acceso al servidor. Un ejemplo de esto es la figura 4.31.

```
 * Debugger is active!
 * Debugger PIN: 204-447-038
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [05/Jul/2021 04:26:52] "GET /consultar/categoria?id_categoria=2 HTTP/1.1" 200 -
127.0.0.1 - - [05/Jul/2021 04:27:34] "GET /consultar/categorias HTTP/1.1" 200 -
```

Figura 4.31. Las peticiones del cliente web fueron procesadas correctamente por la API REST por el protocolo HTTP.

También se valida la ruta que permite eliminar una categoría. La figura 4.32. consulta el elemento a eliminar y la figura 4.33. Válida la eliminación del elemento.

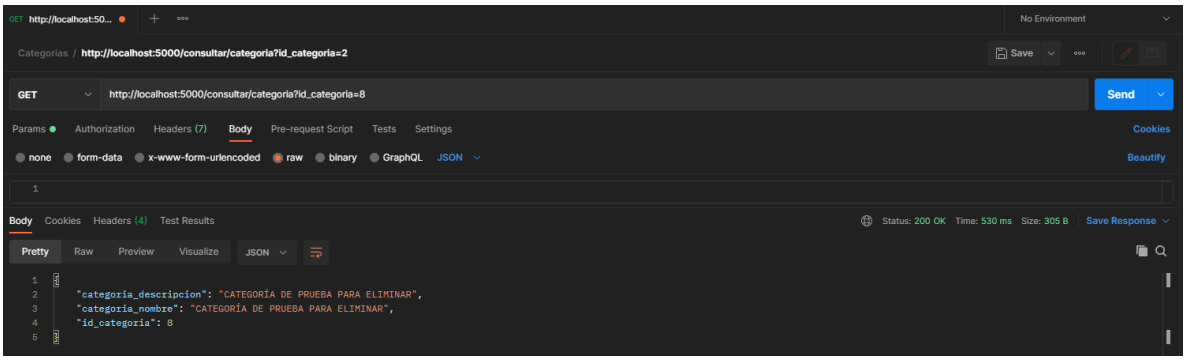


Figura 4.32. GET en POSTMAN para obtener una categoría a eliminar.

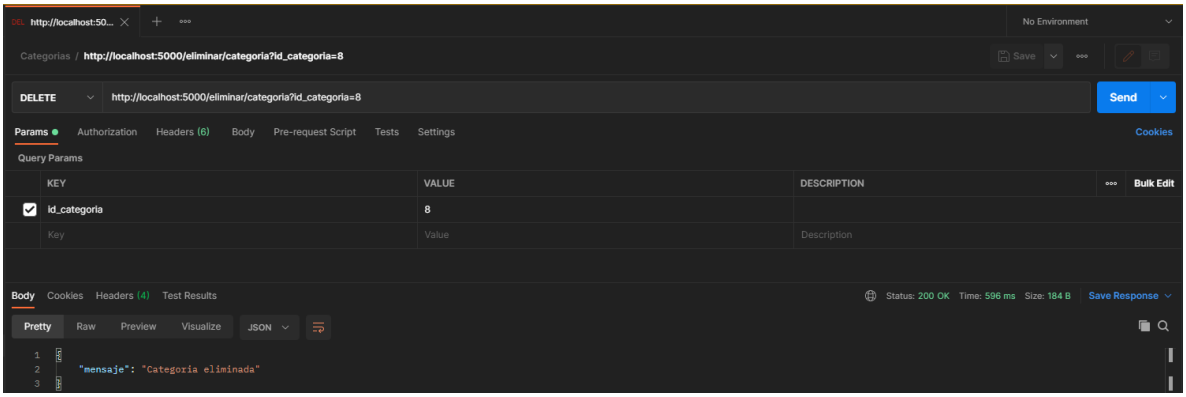


Figura 4.33. DELETE en POSTMAN para eliminar una categoría.

Otras operaciones para POSTMAN es la creación y actualización de un elemento, en la figura 4.34. se crea un elemento con el método POST. En la figura 4.35. POSTMAN va a verificar que el elemento anterior fue creado consultándolo con el método GET. Al verificar que se creó, pasa a actualizar aplicando el método PUT, esto se puede apreciar en la figura 4.36. Y por eso vuelve a consultar el elemento con el método GET para verificar su actualización.

Como se puede observar en las figuras todos los procesos se realizaron correctamente.

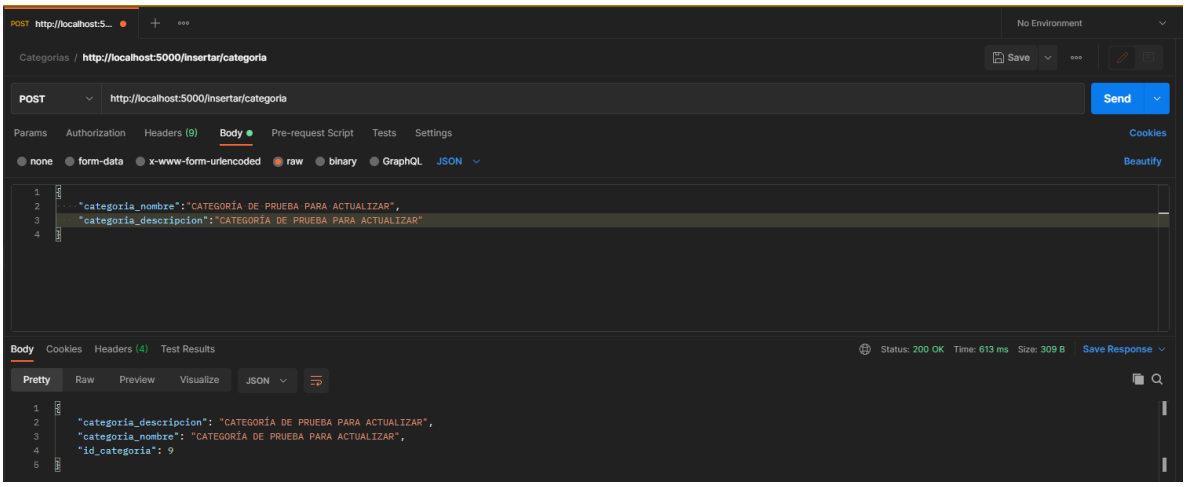


Figura 4.34. POST en POSTMAN para crear una categoría para actualizar.

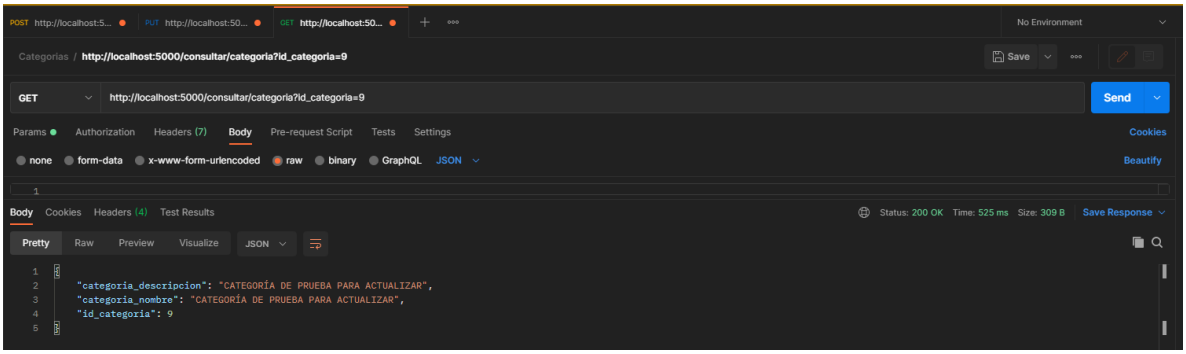


Figura 4.35. GET en POSTMAN para obtener la categoría a actualizar.

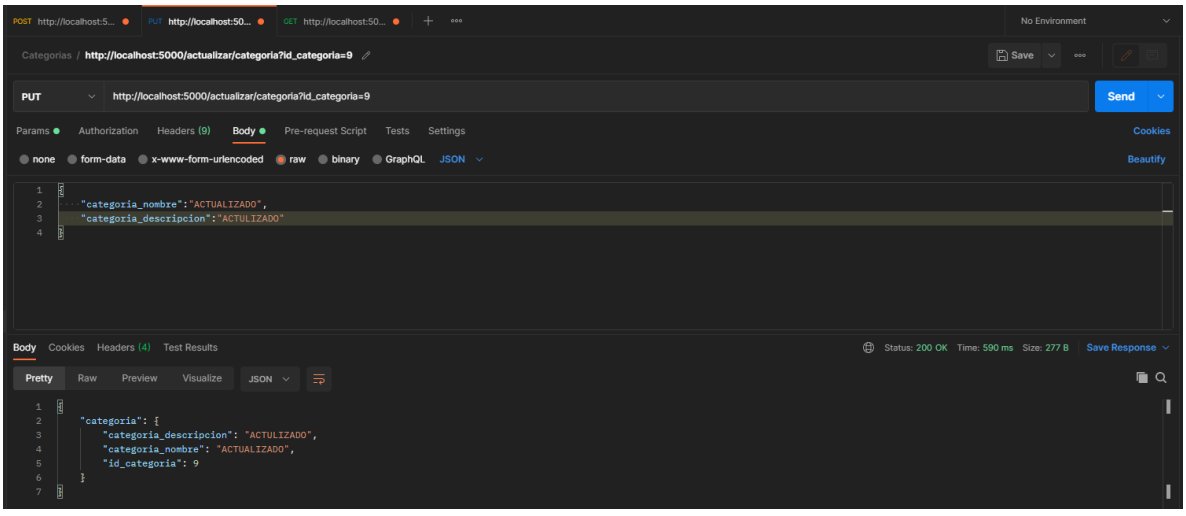


Figura 4.36. PUT en POSTMAN para actualizar una categoría.

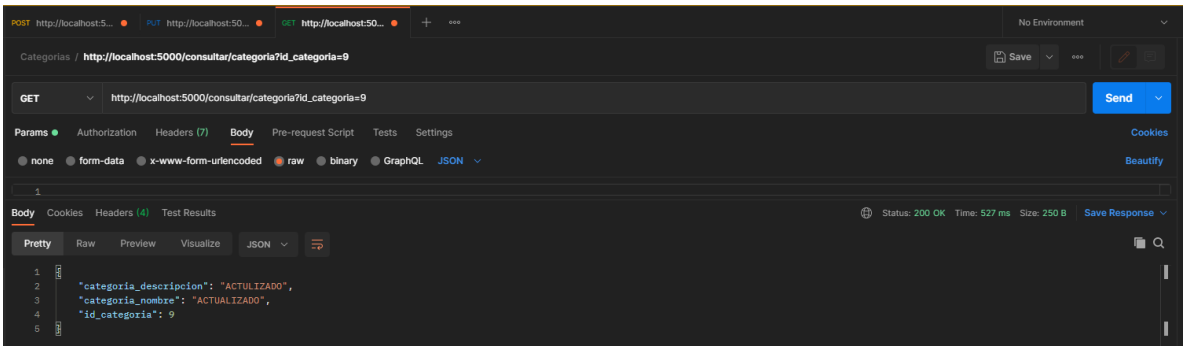


Figura 4.37. GET en POSTMAN para obtener la categoría actualizada.

4.3. Funcionalidad de la aplicación multitáctil multiusuario.

Para iniciar el desarrollo y la implementación de lo previamente diseñado se inicia un proyecto en unity como se muestra en la figura 4.38.

Figura 4.38. Proyecto en UNITY.

Es importante añadir todos los GameObjects necesarios dentro de la escena utilizada. En este caso se utiliza cuatro GameObjects principales o padres con la finalidad de distribuir el proyecto. Esto se puede observar en la figura 4.39.

Figura 4.39. Añadir GameObjects.

Dentro de los GameObjects principales se introducen los GameObjects hijos que tienen como finalidad dar vida al proyecto, son los elementos visuales. Se puede observar cómo fue aplicado en este proyecto en la figura 4.40.

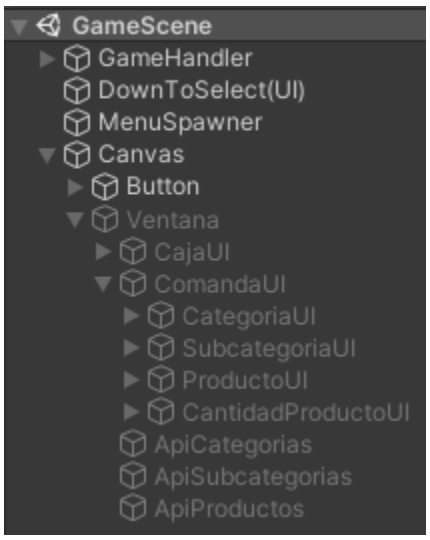


Figura 4.40. Distribución y jerarquía de GameObjects.

En la figura 4.41. se habilita el botón para que el proyecto inicie con el botón que permite generar las ventanas emergentes (Comandas).

Figura 4.41. Habilitar el botón para generar ventanas emergentes.

En la figura 4.42. podemos observar cómo se ve el botón ya habilitado.

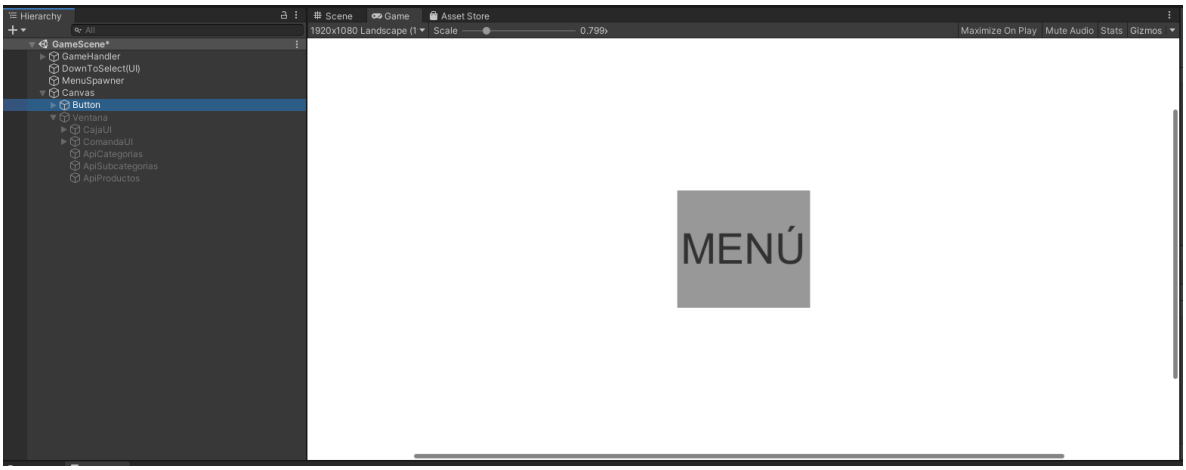


Figura 4.42. GameObject para generar ventanas emergentes.

El proceso que se llevará a cabo para que el comensal pueda realizar su pedido o registro de los productos de interés en su comanda es el siguiente:

- Inicializa la comanda (Abre ventana emergente).
- La comanda inicia con la caja abierta solicitando abrir la comanda.
- Se abre la ventana de categoría donde el comensal puede ver las categorías de productos que el restaurante ofrece.
- Al seleccionar se puede acceder a las subcategorías, los productos o a ambos en caso de que existan productos sin subcategorías y subcategorías a su vez.
- Al seleccionar los productos se despliega la cantidad para añadir a la comanda.
- Al terminar el proceso y de aceptar la comanda, se despliega de nuevo la caja con la posibilidad de añadir más productos a la comanda o a pagar el consumo.

Las ventanas que se involucran en este proceso se pueden ver de la figura 4.43. a la figura 4.47.

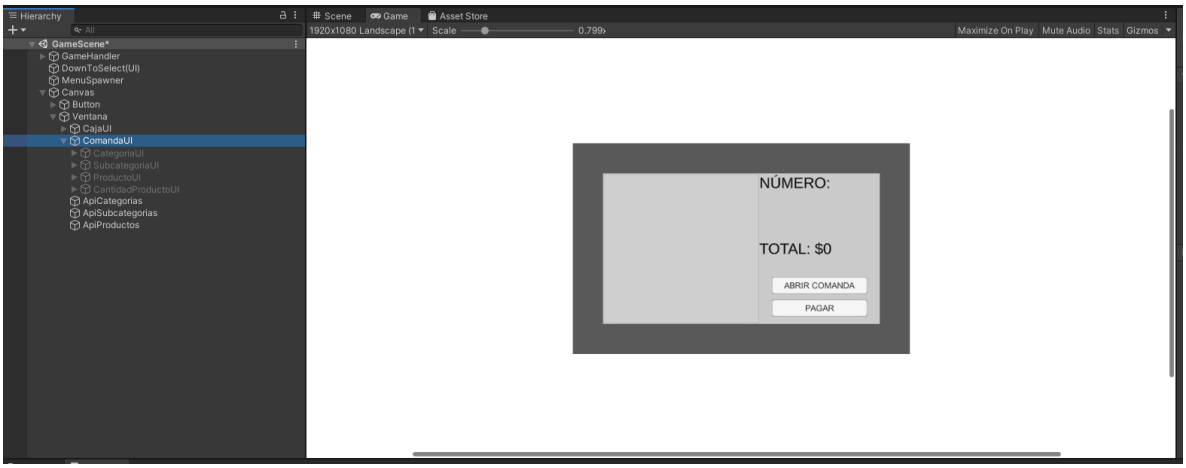


Figura 4.43. Caja UI.

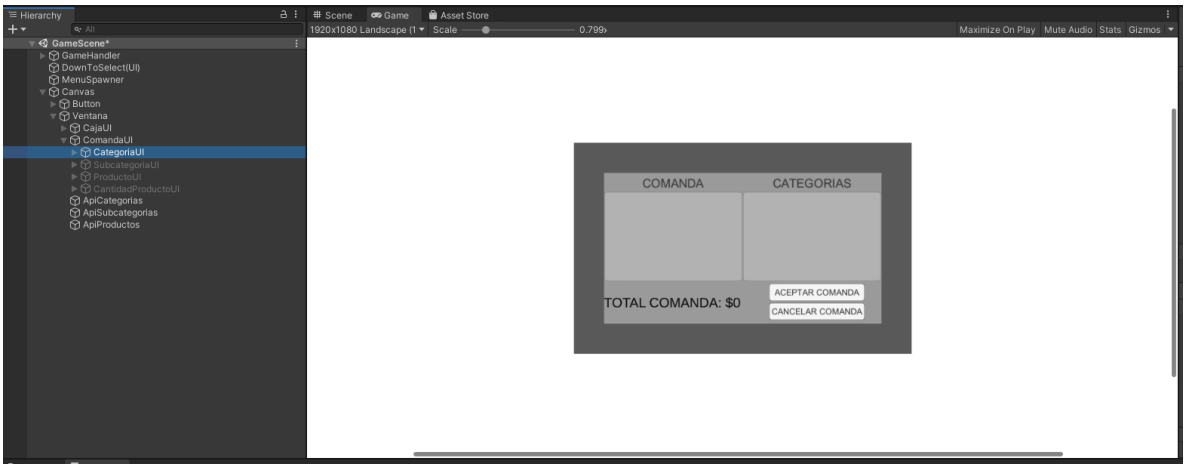


Figura 4.44. Categoría UI.

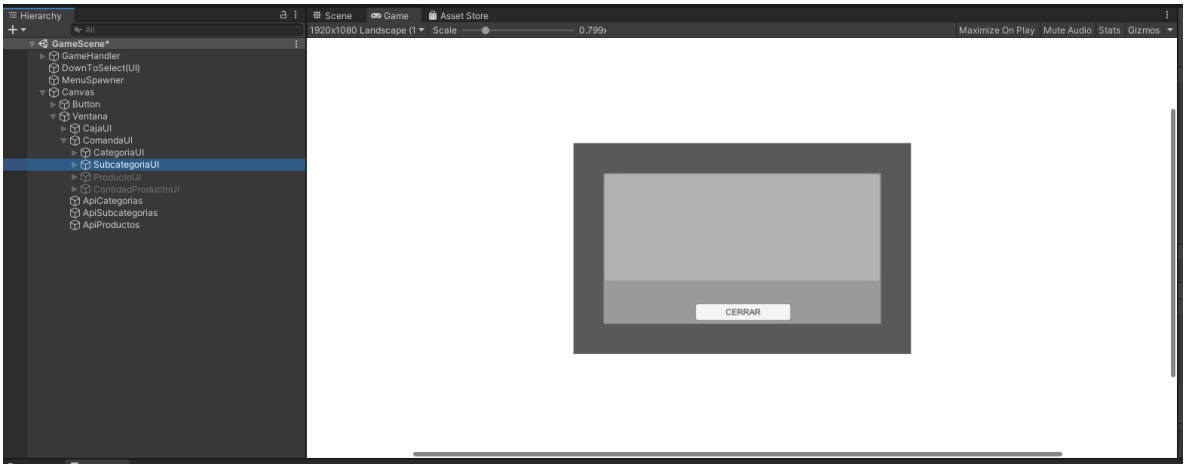


Figura 4.45. Subcategoría UI.

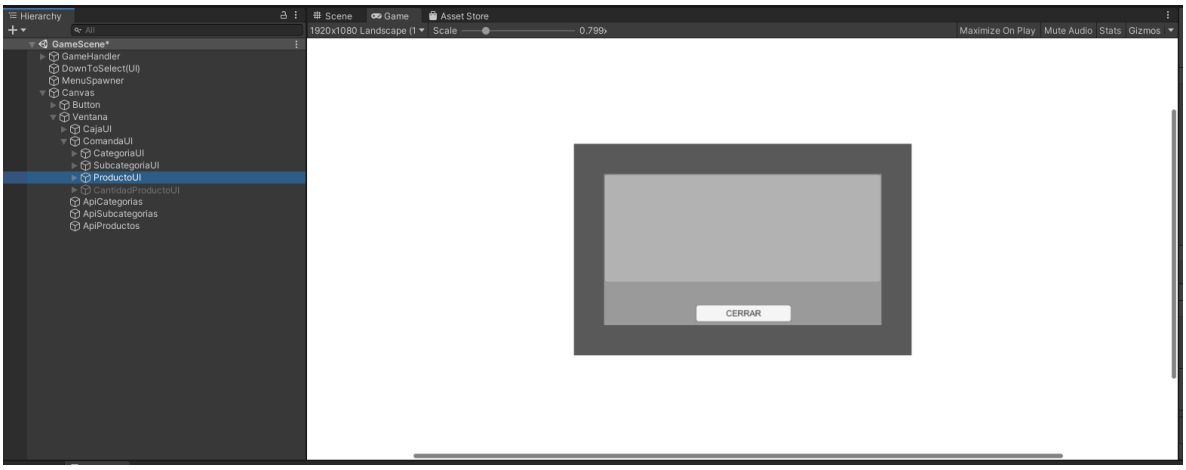


Figura 4.46. Producto UI.

Figura 4.47. Cantidad UI.

Este programa tiene código implementado en cada GameObject a utilizar, sin embargo, la descripción o representación mediante un diagrama es complejo y puede traer a malinterpretar la relación que se puede dar entre clases y GameObjects al momento de asociar los elementos virtuales con cada atributo o método definido en una clase, en caso de requerir el código se recomienda consultar directamente desde el programa de UNITY.

En esta sección se describe el correcto funcionamiento de la aplicación multitáctil multiusuario, al describir el funcionamiento en un documento de investigación no se puede apreciar en su totalidad el funcionamiento real, se recomienda consultar directamente la aplicación desde el programa de UNITY o

desde un dispositivo táctil con acceso a un servidor preparado para interactuar con la aplicación.

En UNITY 3D se inicia el programa, al iniciar se empieza a validar el funcionamiento de la aplicación multitáctil multiusuario, como se puede apreciar en la figura 4.48. Lo primero que se observa es el espacio de interacción limpio, sólo se encuentra el botón que al ser presionado inicia una comanda abriéndola en una ventana emergente.

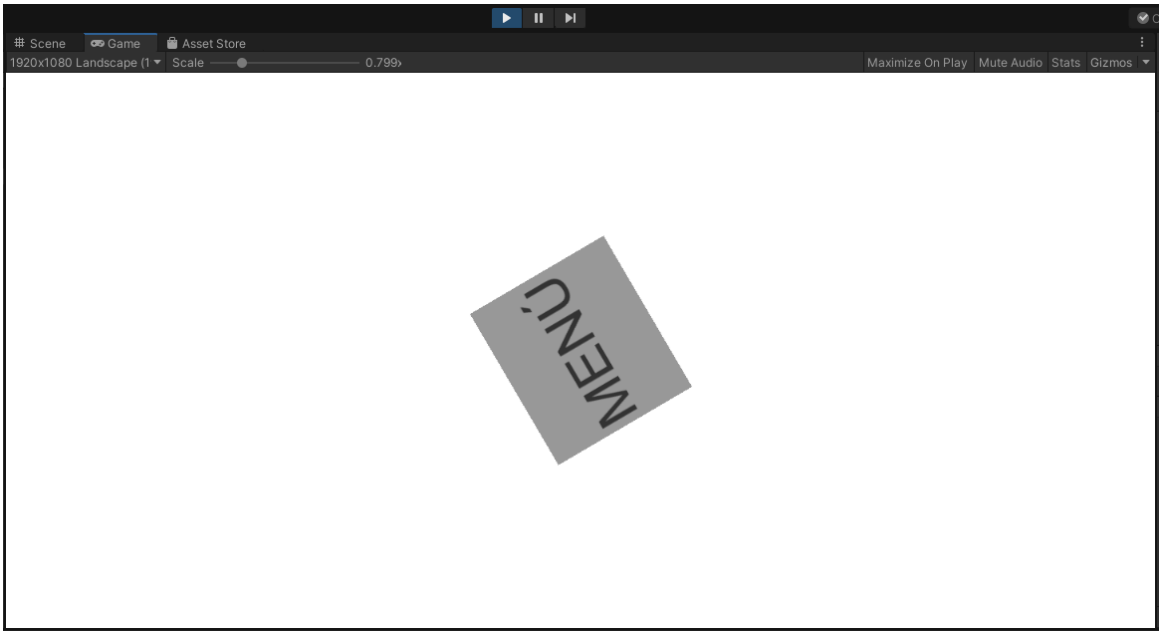


Figura 4.48. Botón para generar ventanas emergentes (Comandas).

Se puede apreciar en la figura 4.49. que al presionar el botón de Menú se despliega una ventana emergente, que permite al usuario iniciar con el registro de sus pedidos. Primeramente, la comanda despliega la interfaz de la caja, esta interfaz permite al usuario o comensal visualizar todos los pedidos realizados durante su estancia en el restaurante y proceder a realizar sus pagos. En este momento la caja se encuentra vacía, si se intenta pagar, no sucederá nada, ya que no cuenta con pedidos, en este caso tendrá que abrir la comanda.

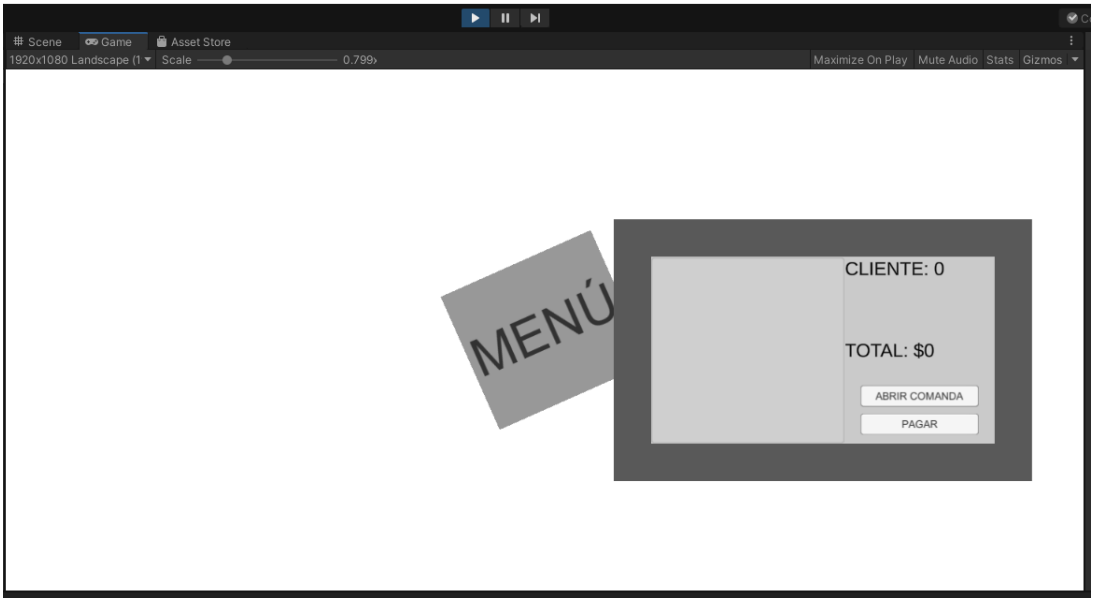


Figura 4.49. Generar la ventana emergente (Comanda).

Al presionar el botón de abrir comanda, se despliega la interfaz de categorías, como se ve en la figura 4.50. el comensal puede visualizar los productos a ingerir.

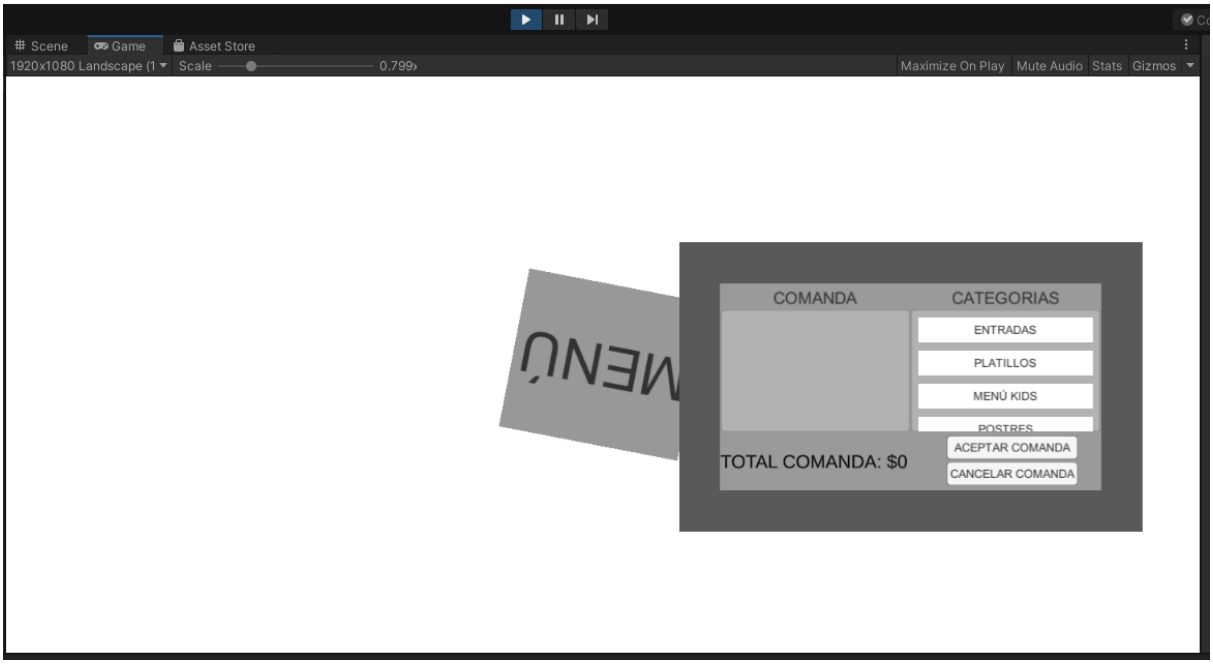


Figura 4.50. Mostrar las categorías.

Al seleccionar una categoría la aplicación muestra los productos como se puede ver en la figura 4.51. Esto sucede porque la categoría seleccionada no cuenta con subproductos y directamente muestra la interfaz de productos.

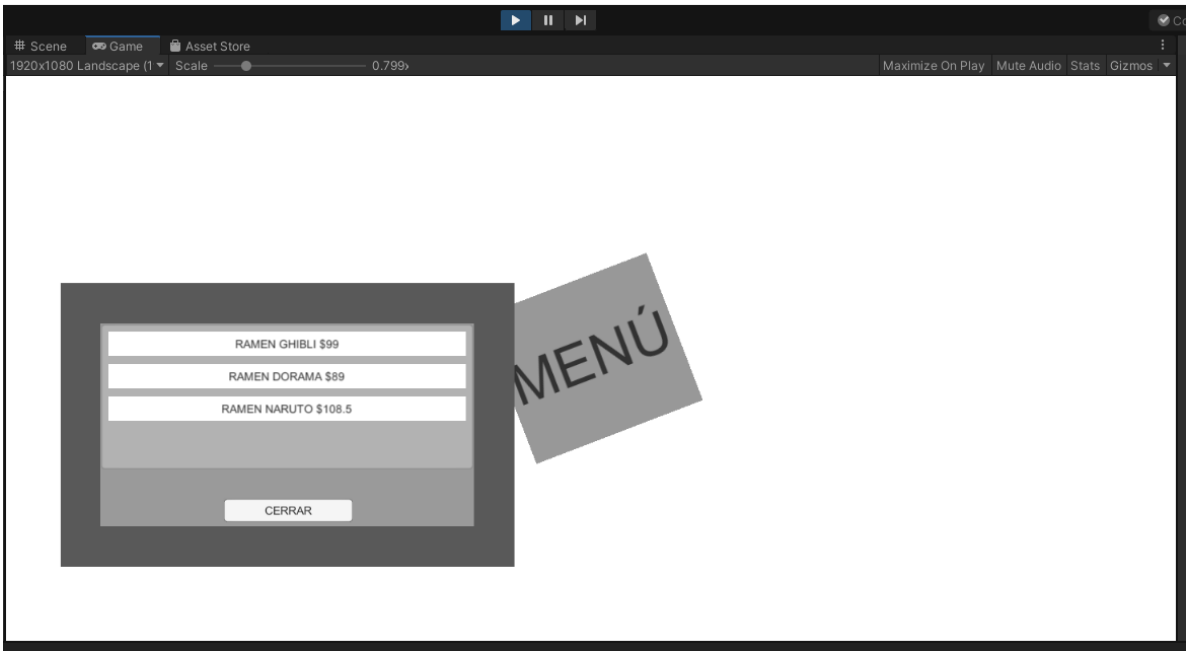


Figura 4.51. Mostrar productos.

Al seleccionar un producto, es porque se desea consumir, entonces se despliega otra interfaz que pregunta al usuario cuantas piezas de dicho elemento va a consumir. Esto se puede consultar en la figura 4.52.

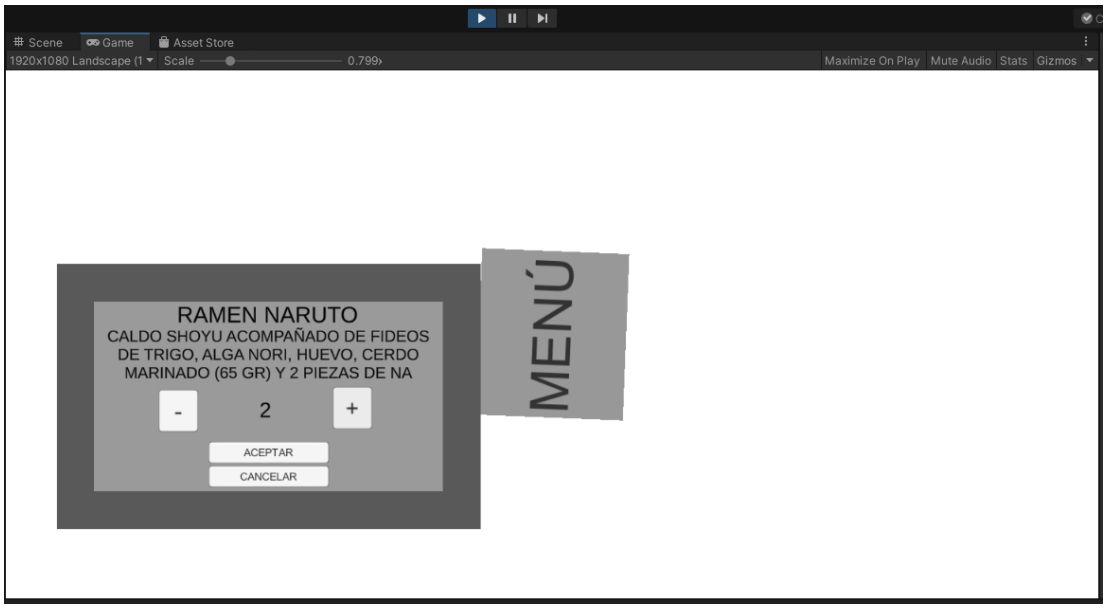


Figura 4.52. Mostrar cantidad de producto a seleccionar.

Al seleccionar productos y seleccionar la cantidad, se empezará a listar dichos productos seleccionados en la sección de comanda, esperarán en una lista hasta que se acepte la comanda, para ser enviados a los trabajadores del restaurante para que esa comanda sea despachada. Se puede observar un ejemplo en la figura 5.53.

Figura 4.53. Generar lista de productos seleccionados (Registrar en la comanda).

Como se observa en la figura 4.54. Puede darse el caso que al seleccionar una categoría ésta cuente con subcategorías, entonces se despliega la interfaz de subcategorías. Adicional pueden existir productos sin subcategorías, entonces ambos elementos compartirán interfaz.

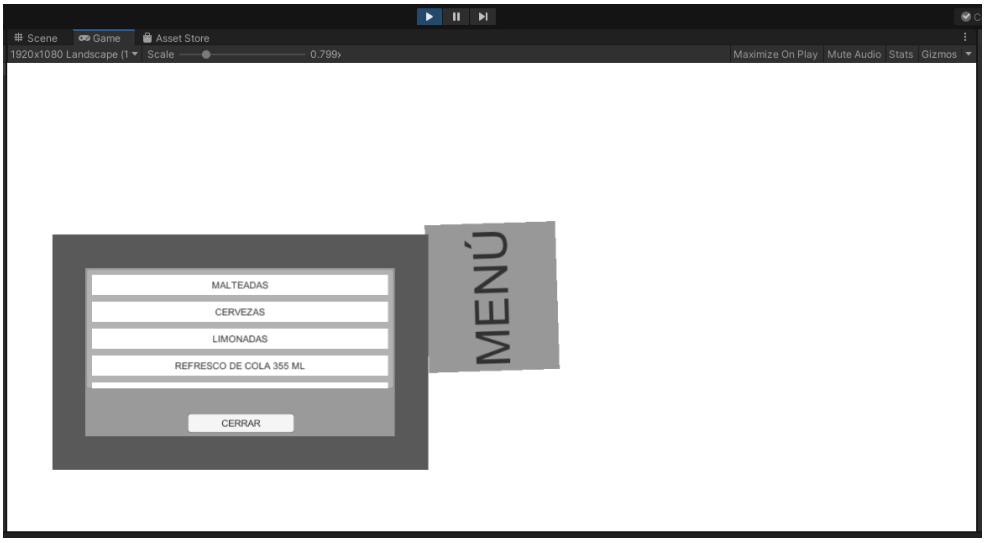


Figura 4.54. Mostrar subcategorías y productos sin subcategorías.

La figura 4.55, muestra más elementos seleccionados y agregados en la lista de la comanda.

Figura 4.55. Registrar varios productos en la comanda.

Estos elementos son botones, por lo que al presionar en ellos estos pueden ser eliminados de la lista de la comanda, si es que el comensal se arrepiente de adquirirlos. Esto se puede observar en la figura 4.56. Y aquí puede suceder dos cosas, la primera que el usuario o comensal cancele la comanda, en este caso no se registra nada y la lista se limpia y la aplicación devuelve a caja al usuario. La segunda e importante es al presionar el botón aceptar comanda. Ya que, al presionar este botón, la lista se manda al restaurante para ser atendido, por lo que el comensal ya solo tiene que esperar en la mesa para recibir la orden.

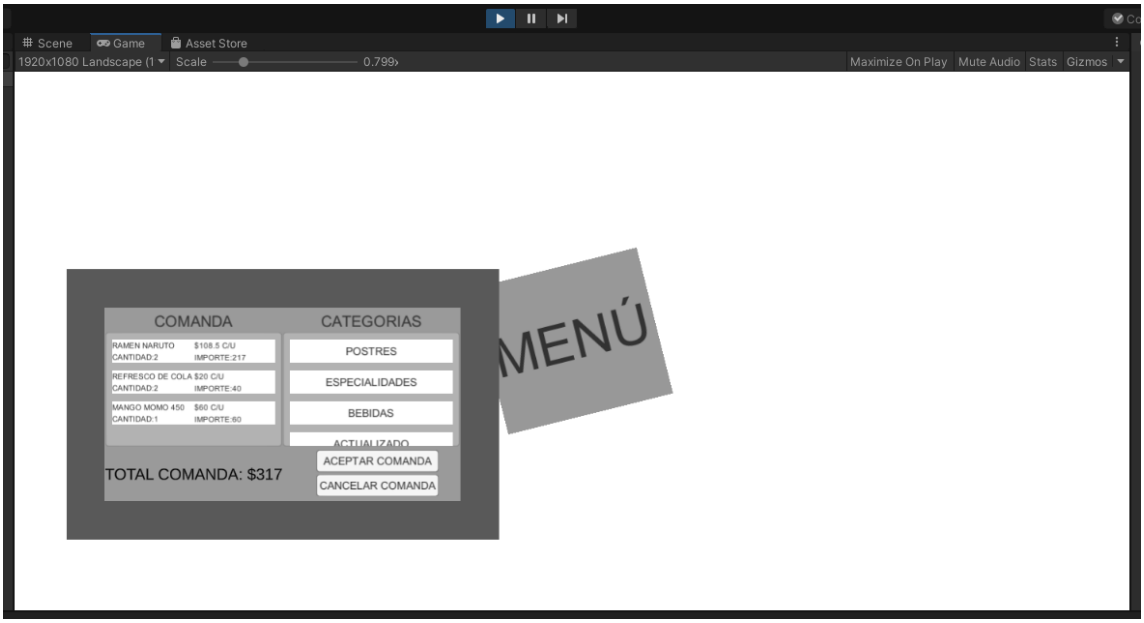


Figura 4.56. Eliminar productos de la comanda.

La figura 4.57. Muestra a la aplicación al aceptar la comanda, como se puede observar los productos aceptados para hacer pedido de ellos se añaden a la lista de la caja, estos elementos ya no pueden ser cancelados, ya que están en preparación. Es posible que algunos productos puedan ser devueltos, sin embargo, la aplicación no cuenta con esa opción.

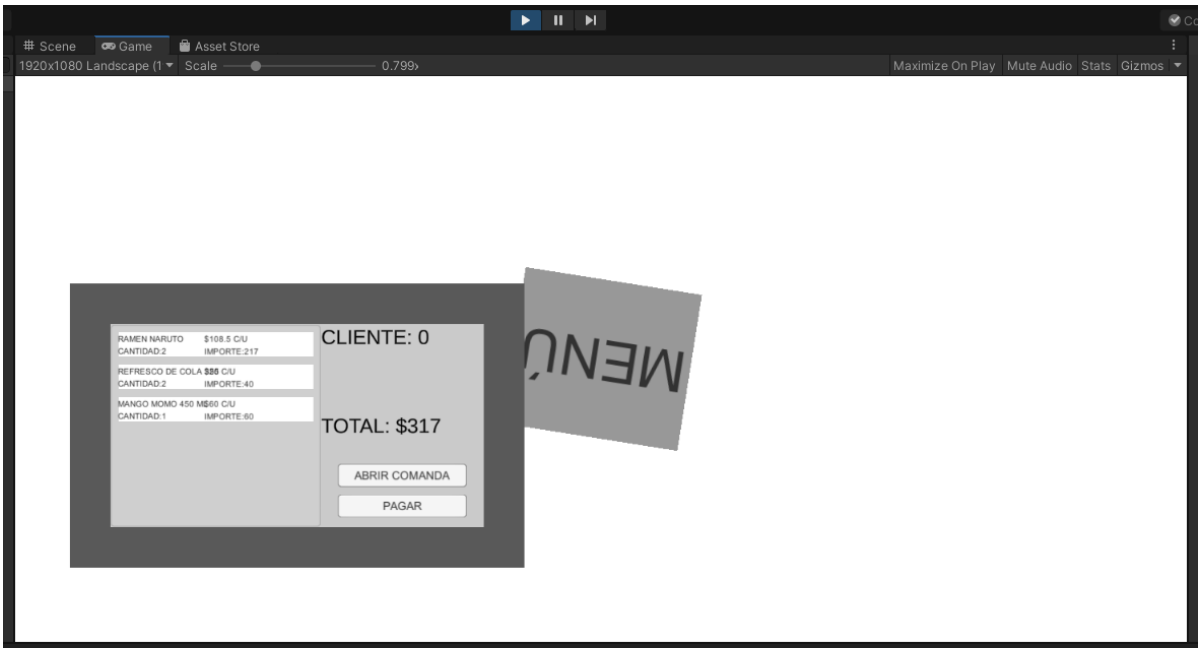


Figura 4.57. Pagar comanda.

Al pagar la comanda, se cierra la ventana emergente (Comanda). Y la información de la compra del comensal durante su estancia en el restaurante es registrada. La idea es que el cliente pueda realizar un pago en efectivo o con tarjeta, pero ese módulo no está integrado en este proyecto de investigación. Esto se refleja en la figura 4.58.

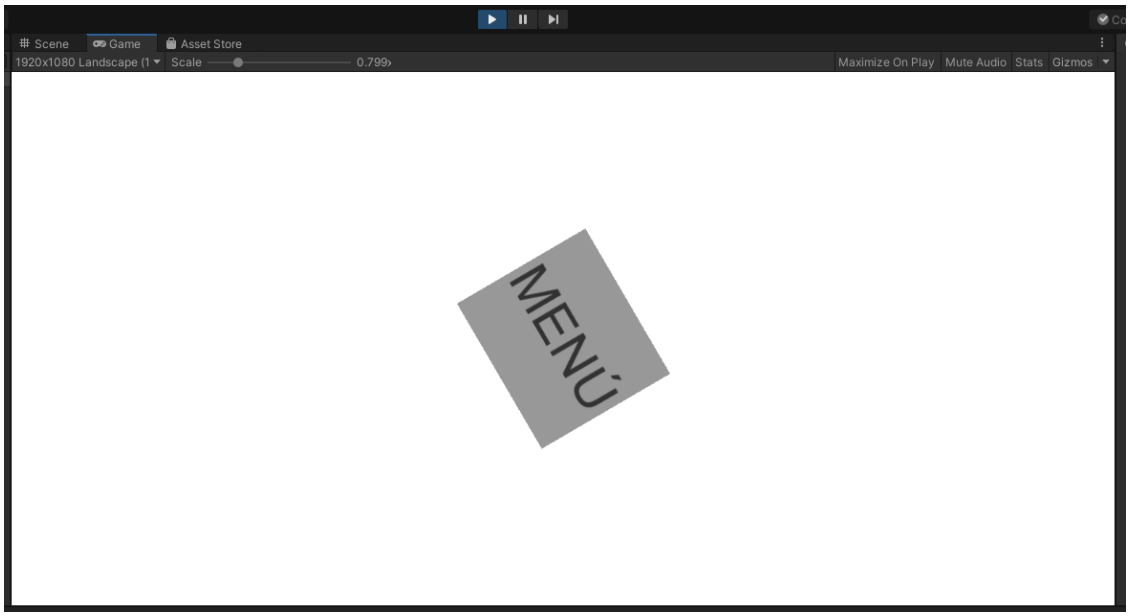


Figura 4.58. Cerrar comanda (Ventana emergente).

Se comprueba que el uso de ventanas emergentes es posible, pero uno de los propósitos de esta investigación es el uso simultáneo de esta aplicación por 4 usuarios. La figura 4.59. muestra que la aplicación desarrollada es capaz de mantener 4 ventanas emergentes abiertas y son totalmente interactivas.

Figura 4.59. Cuatro comandas (Ventanas emergentes) como máximo abiertas simultáneamente.

Como se observa en la figura 4.60. cada una de las ventanas emergentes o comandas virtuales puede interactuar totalmente independiente del resto.

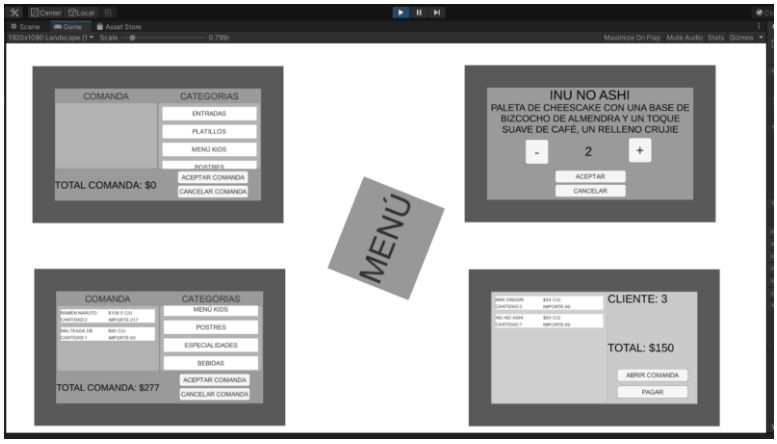


Figura 4.60. Cuatro comandas (Ventanas emergentes) trabajando simultáneamente.

En la figura 4.61. se muestra incluso que una de ellas realiza un pago, sin afectar el proceso del resto.

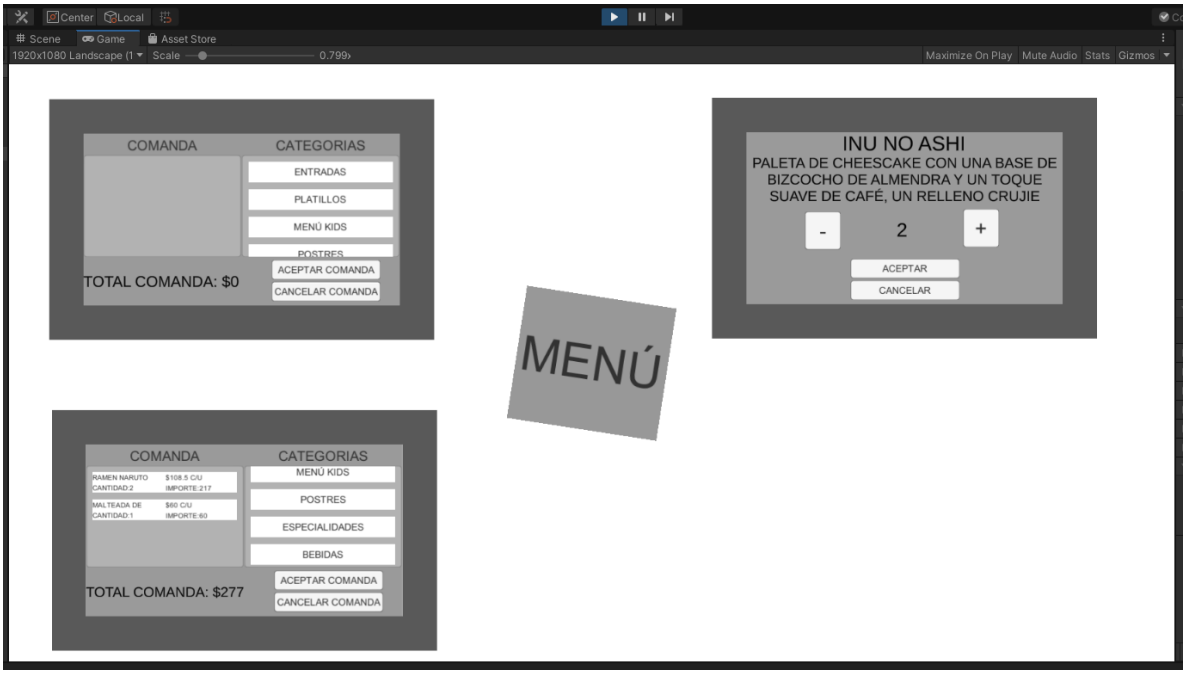


Figura 4.61. La comanda número 3 realizó el pago.

Con POSTMAN se valida que realmente se registró la comanda de la ventana emergente número 3. Esto se puede observar en la figura 4.62.

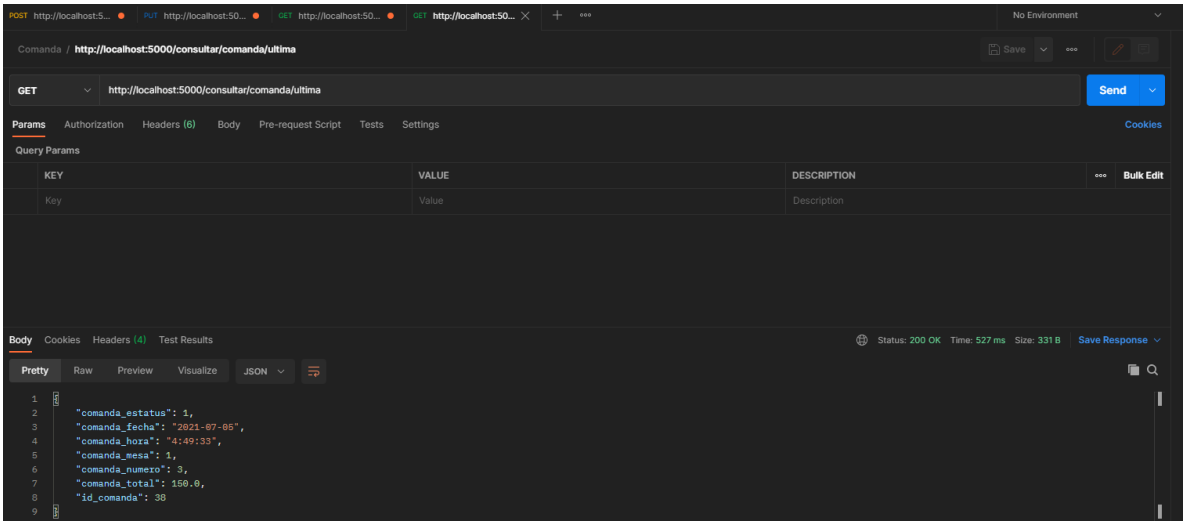


Figura 4.62. Registró del pago realizado por la comanda número 3.

La aplicación funciona correctamente, al comprobar su funcionamiento se obtenido los resultados esperados.

La aplicación despliega una ventana emergente, cada vez que se da un toque al botón que permite abrir una comanda. Dentro de esta ventana se despliega la comanda.

La comanda cuenta con elementos interactivos, como lo es una lista que contendrá el consumo total del comensal y botones que permite interactuar con el menú, contiene los productos que el restaurante va a ofrecer.

El menú que se ofrece contiene, organizará los productos a mostrar en categorías y subcategorías. Al seleccionar cada producto permite seleccionar cuantos productos añadir al pedido que se realiza al momento de seleccionar los productos de interés.

Al terminar de realizar el pedido, se acepta y se manda a registrar a la API. Este pedido es añadido a la lista de la comanda, se registra todos los pedidos que se entiende cómo todo el consumo del comensal en el restaurante.

Lo anterior comentado es un ciclo repetido por cada ventana emergente, desde el momento que es abierta hasta el momento que se cierra al momento de

hacer el pago. Es importante mencionar que la aplicación puede desplegar un máximo de cuatro ventanas emergentes. Tiene la capacidad de desplegar más, por rendimiento se manejó ese límite.

Capítulo 5.

5. Conclusiones y trabajo futuro.

Mediante análisis, desarrollo y diseño se demostró que la aplicación multitáctil multiusuario que fue propuesta permite realmente la interacción multiusuarios en un espacio de trabajo y que es medio para que un comensal o usuario pueda realizar el registro de una comanda en restaurantes, sirviendo esto como autoservicio.

Una limitante importante es, el dispositivo a utilizar, ya que es un factor tal que al ser un dispositivo que no cuente con los requerimientos previamente esperados el funcionamiento de la aplicación puede no ser el mejor.

Además, el costo de una mesa interactiva es muy elevado y se tiene que considerar otros factores, sobre todo económicos para validar el retorno de inversión que pueda generar la implementación de esta tecnología.

Como trabajo futuro se propone una integración de un módulo de pago que permita el realizar transacciones, con la finalidad de cerrar el proceso de compra. Además de mejorar puntos de desarrollo como validaciones, mejorar la interfaz de usuarios, entre otros.

Referencias bibliográficas.

[1] sietemedia. (2009) mesa interactiva, videojuegos. Http://www.sietemedia.com.mx/portfolio/restaurante-ocho-mesa-interactiva/

[2] sietemedia. (2009) muto http://www.sietemedia.com.mx/portfolio/muto-roma-2/

[3] luis a., marcela d. Rodriguez, “interacción humano-computadora y aplicaciones en méxico”

[4] h. Acuña, raul. “interfaces para humanos: más allá de los teclados y ratones.” Chile 2015.

[5] jyotsna j., chetali r., akash s. “dine ‘n’ fine: una guia interactiva” india 2018.

[6] allan c., jonathan b., jiten d., rob e. “interfaces inteligentes re-imaginando la manera como humanos, máquinas y datos interactúan”, deloitte insights, vol .10, pp. 71-73

[7] inegi, “censos económicos 2014 la industria restaurantera en méxico”, inegi, méxico.

[8] joel a. (2018, mayo 28,). News week https://newsweekespanol.com/2018/05/restaurantes-quiebran-captan-esencia-experiencia/

[9] amr (2018, mayo 28,). Amr http://www.amr.org.mx/noticias.phtml?id=2515

[10] vindya l. Achini e., hiranthi p., orabashi m. “foody-smart restaurant y gestion de sistema de pedidos” en ieee 6th region 10 tecnologia humanitaria., diciembre 2018.

[11] alejandro d., federico r., elva e., juan m. “gestión tecnológica en restaurantes: desarrollo y validación de un instrumento de medición.” México, 2017.

[12] muhittin c. “journal of hospitality and tourism technology” usa 2019.

[13] muhittin c. “journal of hospitality and tourism technology” usa 2019.

[14] De Bonis G. (2019, septiembre, 26). germandebonis https://germandebonis.com/la-comanda-de-restaurante-normas-procedimientos-y-ordenes/#%C2%BFQue_es_la_Comanda_de_Restaurante_Guia_de_normas_procedimientos_y_ordenes

[15] interactuando cultura digital, (sin año), “mesa interactiva multitouch profesional”. [internet]. Disponible en: https://www.interactuando.es/totem-interactivos-pantallas/mesa-interactiva-

profesional/#:~:text=la%20mesa%20interactiva%20es%20%e2%80%8b,de%20forma%20simple%20e%20intuitiva

[16] atracsys interactive solutions, *multi-user multi-touch applications guide for professionals*. Suiza: atracsys interactive solutions, 2016, p. 4,5,6,7, 12, 13. Disponible en: https://www.salon-ecom.com/public/exposants_files/multi-user_multi-touch_guide_for_professionnals.pdf

[17] j. R. Oliva haba, m. F. Mate gutiérrez, c. Manjavacas zarco, *montaje y mantenimiento de equipos*. España: paraninfo, 2019, p. 268.

[18] viewsonic, *why touch screen technology?* 2018, p. Portada, 2, 3, 4, 5. Disponible en: https://www.viewsonic.com/touch/touch-screen-technology

[19] lenovo, (sin fecha), "¿qué es la tecnología multi-touch?". [internet]. Disponible en: https://www.lenovo.com/mx/es/faqs/pc-vida-faqs/que-es-la-tecnologia-multi-touch/

[20] arm, (2020), "glossary smart devices". [internet]. Disponible en: https://www.arm.com/glossary/smart-devices#:~:text=smart%20devices%20are%20all%20of,be%20small%2c%20they%20are%20powerful.

[21] fernández m., renukappa s., suresh s. "what is a smart device? – a conceptualisation within the paradigm of the internet of things", springer open, 2018.

[22] chao g. "human-computer interaction: process and principles of human-computer interface design", design & art school, china.

[23] fuentes r., "la carrera de las nuevas interfaces inteligentes", *forbes*, agosto, 2019. [internet]. Disponible en: https://www.forbes.com.mx/la-carrera-de-las-nuevas-interfaces-inteligentes/

[24] bbva, (2017), "¿qué es el interner de las cosas (iot)?". [internet]. Disponible en: https://www.bbva.com/es/internet-las-cosas-iot/

[25] unity, (2020)," plataforma básica de unity". [internet]. Disponible en: https://www.bbva.com/es/internet-las-cosas-iot/

[26] unity, (2020)," plataforma básica de unity". [internet]. Disponible en: https://docs.unity3d.com/manual/index.html?_ga=2.250566594.597051370.1591687428-1803947463.1591586637

[27] openwebinars, (2018), “por qué usar api rest en 2018”. [internet]. Disponible en: https://openwebinars.net/blog/por-que-usar-api-rest/

[28] bbva, (2016), “api rest: qué es y cuáles son sus ventajas en el desarrollo de proyectos”. [internet]. Disponible en: https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos

[29] amodeo e., *principios de diseño de apis rest*. Canada: leanpub, 2013, p. 1.

[30] firefox, (2020), “generalidades del protocolo http”. [internet]. Disponible en: https://developer.mozilla.org/es/docs/web/http/overview

[31] platzi, (2020), “bases de datos desde cero”. [internet]. Disponible en: https://platzi.com/base-de-datos/

[32] oracle, (Sin año), “bases de datos desde cero”. [internet]. Disponible en: https://www.oracle.com/mx/database/what-is-a-relational-database/