# *Diseño de un Prototipo Web Analizador de Datos para el Control y Gestión en la Toma de Decisiones del Restaurante Guilarte de Bogotá.*

**Ing. Anjhelo Giovanny Ramírez Suárez**[1]

**Ing. Andersson Salinas González**[2]

**Facultad Ingeniería, Universidad Distrital Francisco José de Caldas**

**Especialización en Ingeniería de Software**

**Director: MEng. Oswaldo Alberto Romero Villalobos**

**Revisor: MEng. John Jairo Londoño Pérez**

**15 de noviembre de 2022**

[1] Anjhelo Giovanny Ramírez Suárez anramirezs@correo.udistrital.edu.co
[2] Andersson Salinas González asalinasg@correo.udistrital.edu.co

# INTRODUCCIÓN

La analítica de datos es un tema del que se habla bastante, y es que gracias al avance en las ciencias de la información, se pueden crear herramientas de análisis, que permiten encontrar datos escondidos dentro de los que consideramos visibles, es decir, los que notamos en tablas, pero que por detrás contienen información que puede ayudar a optimizar procesos, tomar decisiones, etc.

Al analizar los datos que puede obtener un restaurante teniendo en cuenta el consumo de sus clientes, se pueden descubrir tendencias, se puede saber cuáles son los que más gustan, los que menos, igualmente, al conocer los datos del cliente, se puede conocer su preferencia, lo que permite ejercer un control sobre el menú, aprovechar los que más se venden para promocionarlos, ir reduciendo los que menos para generar otras alternativas, y, en general, tomar mejores decisiones respecto a la administración del negocio, lo que eventualmente derivaría en una mayor rentabilidad.

El objeto del presente proyecto es diseñar un prototipo que permita tomar los datos de las órdenes de servicio del restaurante Guilarte de la ciudad de Bogotá, para analizarlos a través de técnicas estadísticas y ciencia de datos para obtener información que permita mejorar su proceso de toma de decisiones.

# Tabla de Contenido

**INTRODUCCIÓN** **2**

**PARTE I. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN** **6**

**CAPÍTULO 1: DESCRIPCIÓN DE LA INVESTIGACIÓN** **6**

1.1 Planteamiento del problema 6
1.1.2 Formulación del problema 7
1.1.3 Sistematización del problema 7
1.2.1 Objetivo General 7
1.2.2 Objetivos específicos 7
1.3 Justificación del trabajo 8
1.3.1 Justificación metodológica 8
1.3.2 Justificación práctica 8
1.4 Hipótesis 9
1.5 Marco referencial 9
1.5.1 Marco Teórico 9
1.5.2 Marco Conceptual 13
1.5.3 Marco Espacial 14
1.6 Metodología de la investigación 15
1.6.1 Tipo de estudio 15
1.6.2 Método de investigación 15
1.6.3 Fuentes y técnicas para la recolección de la información 15
1.6.4 Tratamiento de la información 15
1.7 Organización del trabajo de grado 16
1.8 Estudio de sistemas previos 16

**PARTE II. DESARROLLO DE LA INVESTIGACIÓN** **17**

**CAPÍTULO 2. ARQUITECTURA EMPRESARIAL** **17**

2.1 Capa de Motivación 17
2.1.1 Punto de Vista de Interesados. 17
2.1.2 Punto de Vista de Realización de objetivos. 17
2.1.3 Punto de Vista de Contribución de objetivos. 18
2.1.4 Punto de Vista de Principios. 19
2.1.5 Punto de vista de la realización de requerimientos. 19
2.1.6 Punto de Vista de Motivación. 20
2.2 Capa de Estrategia 20
2.2.1 Punto de Vista del mapa de capacidad. 20
2.2.2 Punto de Vista de Realización de resultado. 21
2.2.3 Punto de vista Mapa de recurso. 22
2.3 Capa de Negocio 22

2.3.1 Punto de Vista de la Organización. 22
2.3.2 Punto de Vista de Cooperación de Actor 23
2.3.3 Punto de Vista de Función del negocio 23
2.3.4 Punto de Vista de Proceso del negocio 24
2.3.5 Punto de Vista de Cooperación del Proceso de negocio 24
2.3.6 Punto de Vista de Producto. 25
2.4 Capa de Aplicación 25
2.4.1 Punto de Vista Comportamiento de la aplicación. 26
2.4.2 Punto de Vista Cooperación de la aplicación. 26
2.4.3 Punto de Vista Estructura de la aplicación. 27
2.4.4 Punto de Vista Uso de la aplicación. 27
2.5 Capa de Tecnología 28
2.5.1 Punto de vista de la infraestructura. 29
2.5.2 Punto de vista de uso de tecnología. 29
2.5.3 Punto de vista de implementación y organización. 30
2.5.4 Punto de vista de estructura de información. 30
2.5.5 Punto de vista de realización de servicio. 31
2.5.6 Punto de vista de las capas. 31
2.6 Capa de Implementación y despliegue 32
2.6.1 Punto de vista de Proyecto 32
2.6.2 Punto de vista de Migración 32
2.6.3 Punto de vista de Implementación/Migración 33
2.7 Capa Física 34

**CAPÍTULO 3. DISEÑO DEL PROTOTIPO 34**
3.1 Modelo Lógica de Negocio 34
3.2 Modelo Base de datos sugerido 35
3.2.1 Base de datos MySQL 35
3.2.2 Base de datos NoSQL 37
3.3 Arquitectura de software 38

**CAPÍTULO 4. IMPLEMENTACIÓN DEL PROTOTIPO 38**
4.1 Base de datos 38
4.2 Interfaz de usuario 41
4.3 Microservicio Restaurante 42
4.4 Microservicio Analítica de datos 46

**PARTE III. CIERRE DE LA INVESTIGACIÓN 51**

**CAPÍTULO 5. CONCLUSIONES 51**
5.1 Verificación, contraste y evaluación de los objetivos 51
5.1.1. Verificación de Objetivos 51
5.1.2. Evaluación de objetivos. 52
5.2 Síntesis del modelo propuesto 52
5.3 Resultado de Reportes 54
5.4 Aportes originales 58

**CAPÍTULO 6. PROSPECTIVA DEL TRABAJO DE GRADO** **58**
6.1 Alcances y limitaciones 58
6.2 Líneas de investigación futuras 59

**BIBLIOGRAFÍA** **59**

# PARTE I. CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

## CAPÍTULO 1: DESCRIPCIÓN DE LA INVESTIGACIÓN

### 1.1 Planteamiento del problema

Para la realización del planteamiento del problema se tuvieron en cuenta las siguientes variables dependientes que permiten identificar los síntomas del problema.

- Pérdidas económicas.
- Poco control sobre el inventario.
- Toma decisiones equivocadas a la hora de crear los menús diarios.

Estos síntomas tienen causas que en conjunto consiguen generar un diagnóstico al problema de investigación, estas variables independientes son:

- Las pérdidas económicas se deben a que los alimentos perecederos de baja rotación se echan a perder por no conocer las preferencias en cuanto a los menús.
- No existe un sistema automatizado para la gestión de inventarios.
- La toma de decisiones no es certera debido a que no se tiene un registro del histórico de los alimentos con mayor rotación.
- Falta de control sobre el inventario, la toma de decisiones y el análisis de la información.

La toma de decisiones en un restaurante debe tener en cuenta el inventario, los menús y la tendencia de consumo por parte de los clientes, esta toma de decisiones actualmente se apoya de un inventario que se realiza de manera manual con la asistencia de una hoja de cálculo o a papel y lápiz, dando lugar a errores de digitación o tachones e impidiendo la exactitud sobre el control de los productos en el almacén. Al hacerse el abastecimiento del almacén sin tener en cuenta la tendencia de consumo, cabe la posibilidad de adquirir alimentos perecederos de baja rotación lo que causa desperdicios y pérdidas económicas para la empresa.

Si este tipo de situaciones se siguen presentando estas pérdidas podrían crecer exponencialmente y esto podría llevar al cierre de la empresa. Para mitigar las pérdidas se

ha optado por no comprar grandes cantidades de estos alimentos, pero esto hace que suba el precio de los mismos.

Todo este conjunto de problemáticas encontradas ha dado lugar a que el Restaurante Guilarte innove en la toma de decisiones basadas en análisis de datos, y así poder encontrar información que le permita reducir la incertidumbre sobre las variables propias de la inteligencia de negocios.

### *1.1.2 Formulación del problema*

Basados en los datos del negocio, ¿Qué beneficio conlleva el diseño de una herramienta de análisis de datos que a través de los resultados facilite la toma de decisiones respecto a la inteligencia de negocios del restaurante Guilarte?

### *1.1.3 Sistematización del problema*

¿Qué información se debe tener en cuenta para controlar el inventario de un almacén?

¿De qué manera se puede agilizar la toma de decisiones?

¿Cuáles son los indicadores a tener en cuenta para predecir la demanda de consumo?

## 1.2 Objetivos

### *1.2.1 Objetivo General*

Diseñar el prototipo de una herramienta tecnológica que a través de técnicas de análisis de datos permita facilitar la toma de decisiones de acuerdo a la inteligencia de negocios en el restaurante Guilarte de la ciudad de Bogotá.

### *1.2.2 Objetivos específicos*

Sugerir un diseño de base de datos relacional que contenga la información referente a la lógica del negocio para controlar la oferta de productos en el restaurante Guilarte.

Diseñar un algoritmo mediante técnicas de análisis de datos que permita obtener los indicadores de las variables de consumo, costos, gastos y/o productos en stock en el restaurante Guilarte.

Representar los resultados del análisis mediante una interfaz gráfica para sugerir recomendaciones basadas en la tendencia de los datos suministrados por la inteligencia de negocios.

## 1.3 Justificación del trabajo

### *1.3.1 Justificación metodológica*

La razón de ser de la investigación es la de revisar diferentes modelos algorítmicos basados en analítica de datos existentes para la obtención de información implícita en los datos que maneja el restaurante Guilarte, dado que cada uno de los modelos ofrece resultados diferentes de acuerdo a sus valores de entrada, por lo que se requiere la revisión de dichas técnicas y verificar cuál o cuáles de ellas ofrecen respuestas acordes al negocio particular del restaurante Guilarte.

El resultado de la investigación además de implementar el modelo de lógica de negocio para la base de datos del restaurante permitirá realizar una comparativa global de otros modelos de analítica de datos que también se pueden ajustar a la necesidad del negocio para que de esta manera se pueda usar en investigaciones futuras obteniendo una matriz de consulta ajustable a los tipos de datos que se desean procesar.

### *1.3.2 Justificación práctica*

Esta implementación estará enfocada en determinar cómo la analítica de datos puede ser utilizada para mejorar la experiencia de los clientes, también cómo desde la administración del restaurante se puede monetizar el conocimiento de los datos y generar mayor rentabilidad. El resultado más relevante para esta investigación es mostrar la implementación del modelo de inteligencia de negocios donde la información que es susceptible a ser procesada pueda brindar decisiones alternas que harán que el negocio crezca previendo los posibles eventos negativos que impacten el nicho del mismo o prediciendo el volumen de existencias que se necesita en el almacén en un momento específico.

La presente investigación tendrá como resultado el diseño de un prototipo de software que ayude a determinar el beneficio que conlleva el análisis de datos en la toma de decisiones para la inteligencia de negocios, particularmente relacionados con la comida.

## 1.4 Hipótesis

Al implementar una herramienta que permita analizar los datos detrás de los consumos y productos en stock, se evidenciará una mejora en la toma de decisiones sobre la inteligencia de negocios puesto que permitirá generar procesos sobre el cuándo y qué cantidad de insumos debe comprar para mantener abastecido su inventario, así como tomar decisiones respecto a la creación, eliminación de menús para sus comensales.

## 1.5 Marco referencial

### *1.5.1 Marco Teórico*

**1.5.1.1 Administración de restaurantes.** Según (Covarrubias & Garza, 2022) un negocio de la industria gastronómica es de los que más control necesita, ya que al hacerlo de manera pobre, se arriesga su rentabilidad, puede provocar problemas en la cocina, decepciones en los comensales por productos agotados, desperdicio de materia prima, dificultades para mantener los costos de manera adecuada y muchos problemas operativos.

Una de las principales herramientas de control es el inventario, es tan importante que es lo primero que se recomienda preparar, ya que, a través de este se facilita la administración de todos los insumos y su valor económico, evitando algunos de los problemas mencionados anteriormente.

Otra de las herramientas es la comanda, ya que a través de esta, se tiene el control sobre los pedidos de los clientes, lo que junto al inventario constituyen básicamente toda la operación de entrada y salida de los insumos.

Al tener control sobre estas variables, luego, se pueden tomar decisiones sobre lo que se debe comprar y ofrecer alternativas al usuario, además, llegando a recolectar información no solo de lo que se gasta, sino también lo que consumen los clientes, se pueden generar estrategias para mejorar el rendimiento operativo del negocio, al suprimir insumos que no rotan con suficiente frecuencia y permite añadir alternativas con productos que rotan más frecuentemente.

**1.5.1.2 Inteligencia de negocios.** Según (Lee et al., 2021, 2117) La inteligencia de negocios hace referencia al uso de datos para facilitar la toma de decisiones. Los sistemas

de inteligencia de negocios tienen dos procesos que implican la integración (bodega de datos) y el análisis de los datos (OLAP o técnicas de minería de datos). Mediante esta herramienta se extraen los datos de distintas fuentes para luego ser procesados y almacenados en la bodega de datos.

De acuerdo a (Business Intelligence, 2022) algunas de las características de la inteligencia de negocios son:

- Accesibilidad a la información. Teniendo en cuenta que los datos son el activo principal, lo primero, es garantizar el acceso a estos datos sin importar su origen.
- Apoyo en la toma de decisiones. El objetivo es presentar la información de manera que los usuarios puedan extraer aquella que les interese.
- Orientación al usuario final. Busca la independencia del saber técnico y la capacidad para usar herramientas

**1.5.1.3 Analítica de datos.** Según (Swink et al., 2022) La analítica de datos utiliza la información digital con el propósito de extraer conocimiento del análisis para así ayudar en la toma de decisiones.

La analítica está formada por modelos matemáticos, estadísticos y herramientas metodológicas que ayudan a analizar la información y ejecutada por algoritmos que se entrenan para esta meta.

Se puede realizar análisis de la información mediante el análisis predictivo el cual parte de una variable objetiva con la cual se puede medir un indicador y así compararlo con la realidad. También existe el análisis no supervisado, el cual no cuenta con una variable específica y son empleados para segmentar la población.

En el ámbito empresarial, la analitica de datos  permite predecir comportamientos, demanda de productos de una empresa cualquiera, sugerir cuál debe ser el stock en el inventario para atender a esta demanda, teniendo en cuenta factores externos a la organización, como la temporada, las expectativas del consumidor, entre otras variables que pueden lanzar datos (DBA EXPERTS TECH, 2021).

Existen diferentes metodologías para obtener datos susceptibles de ser analizados. Entre las fuentes de información más utilizadas están: Imágenes satelitales, Web tracking, Redes sociales y Archivos de voces.

**1.5.1.4 Arquitectura del Business Intelligence.** Bodega de Datos. Según (Qué Es Un Almacén De Datos, 2022) una bodega de datos también llamada data warehouse, es una metodología de gestión de datos diseñada para habilitar y dar soporte a las tareas de inteligencia empresarial (BI), especialmente las analíticas donde se aplican estadísticas. Los data warehouses solo se han diseñado para realizar consultas y tareas de análisis, y suelen contener grandes cantidades de datos históricos, sin importar cuál sea el uso que se le vaya a dar, cumple con las propiedades como: estabilidad, coherencia, fiabilidad y con información histórica. Puede llegar a tener un gran volumen de datos. El data warehouse presenta las siguientes características según (Kimball & Ross, 2013) :

- Orientado a un tema: La información coleccionada gira en torno a un tema central.
- Integrado: Incluye datos consistentes de varias fuentes.
- Variable en el tiempo: Los datos se basan en fechas o hechos, por lo que pueden variar.

Las bodegas de datos brindan apoyo en el proceso de toma de decisiones de la administración, además de estar orientada al manejo de grandes volúmenes diferentes de datos y que provienen de múltiples fuentes.

Datamart. De acuerdo a (Conexión ESAN, 2019) un Data Mart es un almacén de datos orientado a un área específica, como por ejemplo, área Comercial, Talento Humano u otros sectores en una organización. Por ello, también se le conoce como una base de información departamental. Se caracteriza por disponer la estructura óptima de datos para analizar la información al detalle desde todas las perspectivas que afecten a los procesos de dicha área. Un datamart puede ser alimentado desde los datos de una bodega de datos, o integrar de manera independiente una recopilación de distintas fuentes de información.

Procesamiento analítico en línea. (Microsoft, 2022) menciona que el procesamiento analítico en línea, es a la vez un concepto y una tecnología que tiene como fin proporcionar

un análisis basado en grandes bases de datos, se diseñó para ayudar a obtener datos empresariales con un alto rendimiento, además, estas bases de datos se encuentran optimizadas para grandes cantidades de trabajo de lectura y pocas de escritura. Permiten describir el comportamiento del negocio, basados en acontecimientos que ya han sucedido.

Big Data. (FundéuRAE, 2013) define a esta locución como el conjunto de datos (o dataset) que superan la capacidad de captura, gestión y procesamiento en un tiempo determinado, superando la capacidad del software habitual. Además de la captura, gestión y procesamiento, las dificultades más comunes son también la búsqueda, compartición, visualización, el almacenamiento y análisis, entre otras. Esta tendencia a manejar gran cantidad de datos se debe en muchos casos, a análisis o disciplinas en las que se necesita relacionar datos como puede ser una investigación o datos de enfermedades infecciosas, meteorología, genómica, análisis de negocios, etc. (Diccionario Actual, 2022) menciona que también se puede definir a esta frase como el análisis y tratamiento de grandes repositorios de datos, tan enormes que es imposible tratarlos con las herramientas de analíticas y bases de datos convencionales. Esto se inserta en un entorno como la proliferación de páginas web, aplicaciones de videos e imágenes, redes sociales, dispositivos móviles, sensores, etc. IBM considera que el manejo y tráfico de todo esto es capaz de generar más de 2.5 quintillones de bytes al día. Esta empresa también ha dicho que el universo de Big Data se explica a partir del volumen, variedad y velocidad y por eso consideran que hay un gran desafío hacia adelante, ya que después del 2020 es muy probable que haya 35 zettabits de información. Agregan además que ya no es suficiente el análisis de datos tradicional.

ETL (Extract, Transform & Load Processing). Es el proceso mediante el cual se extraen los datos desde un sistema origen, se transforman y finalmente se cargan en un destino para ser almacenados. Dependiendo del sistema origen, puede o no ser necesario el desarrollo de aplicaciones a medida que realicen esta labor. Esto en razón que es común encontrar sistemas antiguos que definen sus propios lenguajes y protocolos. Por el contrario, “en aplicaciones más recientes es común el uso de lenguajes de programación

más ampliamente utilizados y mutuamente compatibles, así como estándares en el almacenamiento y consulta de bases de datos con lo cual el proceso de extracción puede ser mucho más transparente y fácilmente realizable". A partir de los sistemas operacionales de una compañía, es importante entender de construcción del ETL (Extracción, Transformación y Carga) (Martinez Garcia, 2010):

- Extracción: Proceso de obtención de información de las distintas fuentes.
- Transformación: Filtrado, limpieza, depuración, homogeneización y agrupación de información. la información.

### *1.5.2 Marco Conceptual*

**1.5.2.1 Aprendizaje automático.** Esta subespecialidad de la informática (Russell et al., 2020) se engloba en el campo de la inteligencia artificial y es conocida también como machine learning. Está relacionada con el diseño y desarrollo de algoritmos que permiten a las computadoras promover la acción en base a datos empíricos. Su objetivo es aprender a reconocer automáticamente patrones complejos y tomar decisiones inteligentes basadas en datos. El procesamiento del lenguaje natural es un ejemplo de aprendizaje automático (Emérita Legal, 2018).

**1.5.2.2 Redes neuronales.** Esta técnica de análisis de datos consta de modelos computacionales basados en estructura y funcionamiento de redes biológicas. Al igual que células y conexiones trabajan dentro del cerebro, estas redes permiten encontrar patrones. Se especializan en los no lineales y se recomiendan para aplicaciones que implican tanto aprendizaje supervisado como no supervisado. Un ejemplo de esta clase de técnicas sería la identificación de clientes en riesgo de abandono (10 Técnicas De análisis De Datos Para Estadísticas De Big Data, 2017).

**1.5.2.3 Aprendizaje de reglas de asociación.** Se trata de un conjunto de técnicas de análisis de datos empleadas para descubrir relaciones interesantes entre variables en grandes bases de datos (Menzies & Hu, 2003). La generación y prueba de posibles reglas es el resultado de la aplicación de algoritmos y, en la práctica, uno de sus usos más comunes es el análisis de la cesta de la compra para sugerir recomendaciones, que permite

a los minoristas determinar qué productos se compran con mayor y menor frecuencia para optimizar su planificación y decisiones de abastecimiento (González, 2021).

De acuerdo a (Rodrigo, 2018), se define como una implicación del tipo “si X, entonces Y” (X ⇒ Y), donde X e Y son itemsets o ítems individuales. El lado izquierdo de la regla recibe el nombre de antecedente, y el lado derecho, se denomina consecuente. Se utilizan dos reglas significativas denominadas “soporte y confianza”, la primera indica la frecuencia de un elemento en todos los datos y el segundo indica la probabilidad del consecuente cuando ha sucedido el antecedente.

**1.5.2.4 Algoritmos genéticos.** Desde su aparición en la década de los 60, los algoritmos genéticos han ganado popularidad gracias al crecimiento de la capacidad computacional, son estrategias basadas en la naturaleza, específicamente, vienen de la teoría de la selección natural de Charles Darwin. Al aplicar esta técnica, las soluciones potenciales se codifican para ser combinadas entre sí e incluso someterse a mutaciones, igual que podría hacerse en un laboratorio con cromosomas. Así, tras ser estudiados, los cromosomas individuales se seleccionan para la supervivencia dentro de un entorno modelado que determina la aptitud o el rendimiento de cada uno en comparación con el resto de la población. Los algoritmos genéticos se suelen emplear para fines tan dispares como optimizar el rendimiento de una cartera de inversiones o para mejorar la programación de trabajos en procesos de fabricación (Gutiérrez Reina & Rodríguez del Nozal, 2020).

**1.5.2.5 Análisis de series temporales.** En este caso, se analizan secuencias de puntos de datos, que representan valores en sucesivos tiempos, para extraer las características más significativas de la información. Podría decirse que se trata del uso de un modelo para predecir valores futuros de una serie temporal en base a valores pasados conocidos de la misma u otras series. El pronóstico de cifras de ventas sería una de sus aplicaciones en la empresa (Peña & De Rivera, 2010).

### *1.5.3 Marco Espacial*

El presente proyecto se desarrolló dentro del restaurante Guilarte ubicado en la ciudad de Bogotá.

## 1.6 Metodología de la investigación

### *1.6.1 Tipo de estudio*

Esta investigación se trabajó tipo descriptiva puesto que a través de la identificación de las variables relacionadas al negocio del restaurante Guilarte respecto a su inventario e inteligencia de negocios, su posterior recolección derivará en un análisis de toda esta información que permitirá la generación de modelos que la administración del restaurante podrá analizar para la definición de estrategias y procesos de toma de decisiones respecto al inventario y creación de menús.

### *1.6.2 Método de investigación*

A partir de la identificación de las variables mencionadas con anterioridad, y la respectiva implementación de las consultas, se presentan en forma gráfica a la administración del restaurante para que pueda generar estrategias y tomar decisiones, partiendo de esta premisa, el método de investigación es de tipo deductivo.

### *1.6.3 Fuentes y técnicas para la recolección de la información*

En primer lugar, la información con la que se alimenta la base de datos a analizar corresponde a información proporcionada por el restaurante Guilarte, es decir, por su administración, para esto, se acudió a fuentes primarias a través de una serie de entrevistas con preguntas relacionadas con los siguientes temas: tendencias de consumo, cantidad de veces que han respondido que no pueden servir determinado plato, cantidad de veces que han tenido que preparar platos especiales con insumos del inventario que rotan muy poco, entre otras preguntas.

Ahora bien, para su respectivo análisis, es indispensable el uso de fuentes secundarias como artículos sobre modelos y técnicas de análisis de datos, definición de estrategias y toma de decisiones empresariales, etc.

### *1.6.4 Tratamiento de la información*

La información recolectada se representó en forma de tablas, se ejecutaron consultas a la base de datos y se generaron reportes, el resultado de dicho análisis se

presenta en forma de reportes y gráficos para que el personal administrativo lo interprete y tome decisiones basado en dicha información.

### 1.7 Organización del trabajo de grado

El presente trabajo se divide en 3 partes principales: Contextualización, Desarrollo, Cierre, a su vez, cada una de ellas, posee unos capítulos que describen la metodología utilizada, a continuación, una breve descripción

***Contextualización:*** Hace referencia a la definición del problema de investigación, de los objetivos, a la metodología y a la justificación del mismo.

***Desarrollo:*** En esta parte, se expone la arquitectura empresarial, el modelo de negocio definido, las herramientas técnicas utilizadas para finalmente implementar el prototipo.

***Cierre:*** Esta última parte, hace referencia a la descripción de los resultados obtenidos, al contraste de la hipótesis con los mismos y las conclusiones finales del proyecto.

### 1.8 Estudio de sistemas previos

***Toteat*** (*Toteat Sales Manager*, 2022) es una plataforma de administración de delivery que ofrece soluciones a negocios como restaurantes, cafeterías, bares, almacenes, licorerías, entre otras, ofrecen una herramienta para la gestión de negocios que incluye reportería en tiempo real, reportes relacionados con ventas diarias, inventarios, propinas, entre otras, así como integraciones para otro tipo de servicios como: Domicilios *(Rappi, Uber eats)*, Pagos en línea *(Mercado pago)*, Facturación electrónica *(Factura 1)*.

***PirPos*** (*PirPos*, 2022) es un software para la administración de negocios gastronómicos que permite la toma de órdenes y servicio de mesas, gestionar el movimiento de inventario, y generar estadísticas con reportería útil para informes contables. PirPos, además, ofrece herramientas de integración para pasarelas de pagos.

# PARTE II. DESARROLLO DE LA INVESTIGACIÓN

# CAPÍTULO 2. ARQUITECTURA EMPRESARIAL

## 2.1 Capa de Motivación

Esta capa destaca los diferentes aspectos motivacionales que denotan principalmente aspectos organizacionales, en términos de cada una de las entidades encontradas y sus interacciones. Entidades como objetivos y principios que determinan de forma detallada aspectos fundamentales de la estructura organizacional *(Objetivos, Plan estratégico, Misión y Visión)*

A continuación se presentan cada uno de los puntos de vista de la capa de Motivación a partir del soporte realizado por los interesados para el análisis de datos del restaurante Guilarte de Bogotá.

### *2.1.1 Punto de Vista de Interesados.*

El punto de vista de los interesados permite hacer un acercamiento del modelo de los interesados, donde se pueden describir los objetivos iniciales de alto nivel. A continuación, en la *Figura 1* se plantea el Caso para el Punto de Vista de Interesados con cada uno de los elementos que interactúan entre sí.

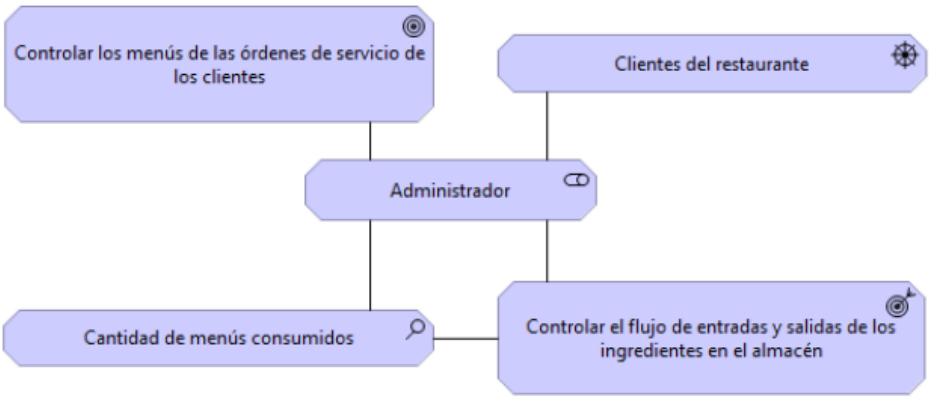



*Figura 1. Capa de Motivación, Punto de vista de los interesados. (Elaboración propia)*

### *2.1.2 Punto de Vista de Realización de objetivos.*

El punto de vista de realización de objetivos permite modelar el refinamiento de metas (de alto nivel) en metas más concretas y el refinamiento de objetivos concretos en requisitos o restricciones que describen las propiedades que se necesitan para realizar las

metas. En la *Figura 2* se puede apreciar el punto de vista de la realización de los objetivos desde la capa de motivación.

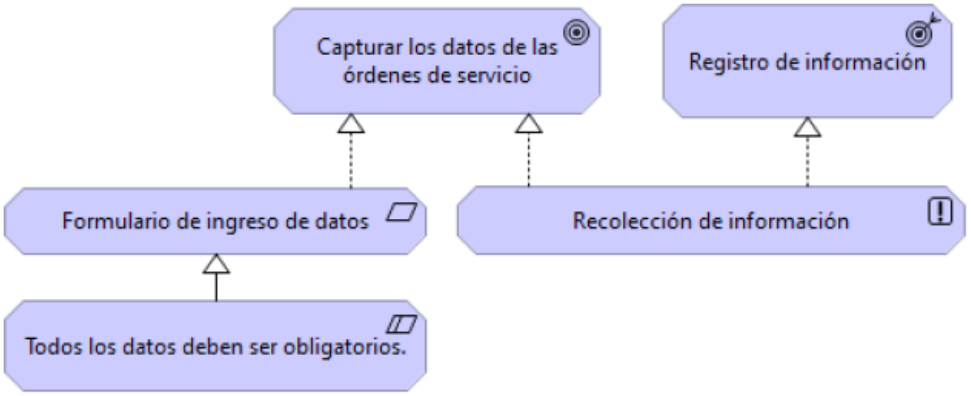



*Figura 2. Capa de Motivación, Punto de vista de la realización de los objetivos. (Elaboración propia)*

### *2.1.3 Punto de Vista de Contribución de objetivos.*

El Punto de Vista de Contribución de Objetivos permite modelar las relaciones de influencia entre objetivos y requisitos. Las vistas resultantes en este punto de vista pueden usarse para analizar el impacto que las metas tienen entre sí o para detectar conflictos entre los objetivos de las partes interesadas.

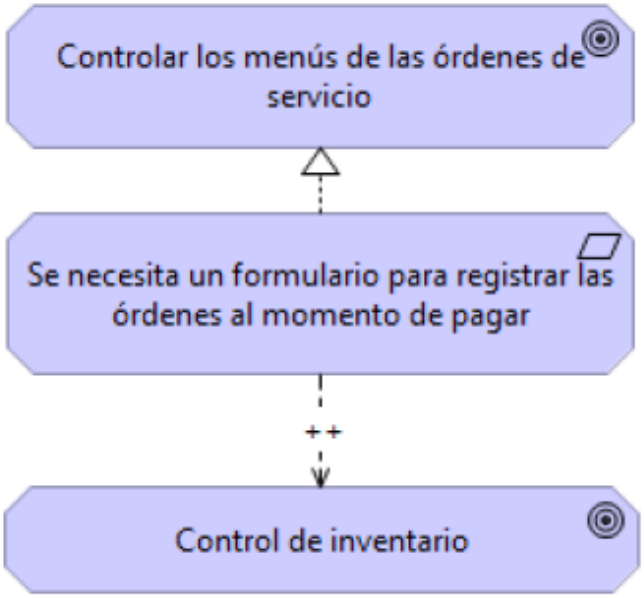



*Figura 3. Capa de Motivación, Punto de vista de la contribución de los objetivos. (Elaboración propia)*

### *2.1.4 Punto de Vista de Principios.*

El Punto de Vista de Principios permite al analista modelar los principios que son relevantes para el problema de diseño en cuestión, incluyendo los objetivos que motivan dichos principios

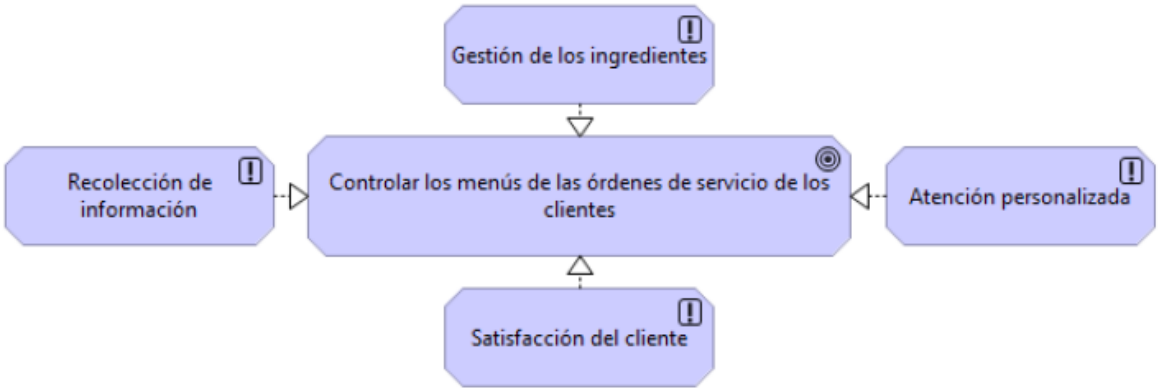



*Figura 4. Capa de Motivación, Punto de vista de Principios. (Elaboración propia)*

### *2.1.5 Punto de vista de la realización de requerimientos.*

Este punto de vista permite a los implicados modelar la realización de los requisitos por parte de los elementos básicos, como los actores empresariales y los servicios empresariales. Este punto de vista tiene en cuenta elementos como los procesos empresariales, los servicios y componentes de la aplicación. Además, puede usarse para refinar requisitos en requisitos más detallados.

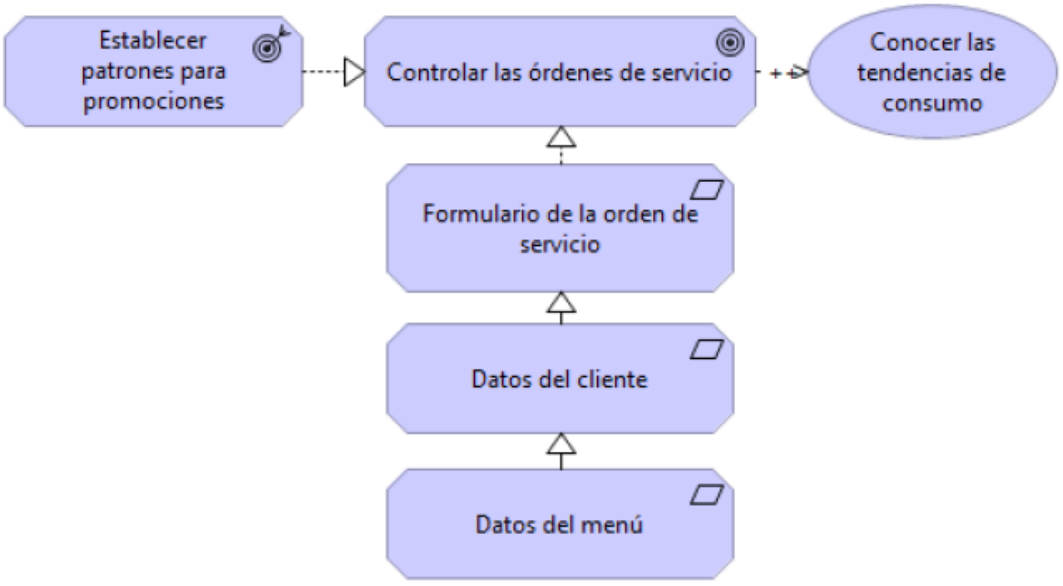



*Figura 5. Capa de Motivación, Punto de vista de Realización de requerimientos. (Elaboración propia)*

### *2.1.6 Punto de Vista de Motivación.*

El Punto de Vista de Motivación permite al diseñador o analista modelar el aspecto de las razones (motivaciones), que guían el diseño o el cambio de una Arquitectura Empresarial. Este punto de vista puede utilizarse para presentar un panorama completo o parcial del aspecto de la motivación relacionando a las partes interesadas, sus objetivos principales, los principios que se aplican y los principales requisitos de servicios, procesos, aplicaciones y objetos.

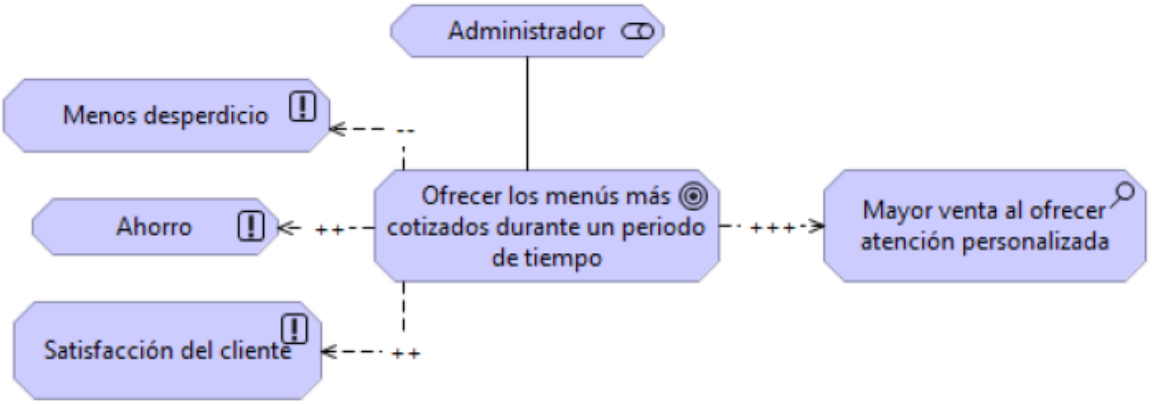



*Figura 6. Capa de Motivación, Punto de vista de Motivación. (Elaboración propia)*

## 2.2 Capa de Estrategia

Esta capa plantea los aspectos estratégicos de la empresa. Cada uno de los puntos de vista que contiene esta capa presenta una punto de vista diferente sobre el modelado de la dirección estratégica de alto nivel y la composición de la organización.

### *2.2.1 Punto de Vista del mapa de capacidad.*

El punto de vista del mapa de capacidad permite crear una visión general estructurada de las capacidades de la empresa. Un mapa de capacidades generalmente muestra dos o tres niveles de las capacidades en toda la organización.

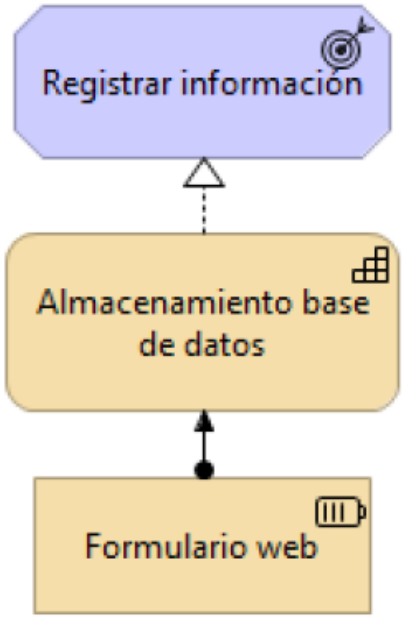



*Figura 7. Capa de Estrategia, Punto de vista de Mapa de capacidad. (Elaboración propia)*

### *2.2.2 Punto de Vista de Realización de resultado.*

El punto de vista de realización del resultado se utiliza para mostrar los resultados orientados al negocio, considerados de alto nivel, son realizados por las capacidades y los elementos centrales de la organización.

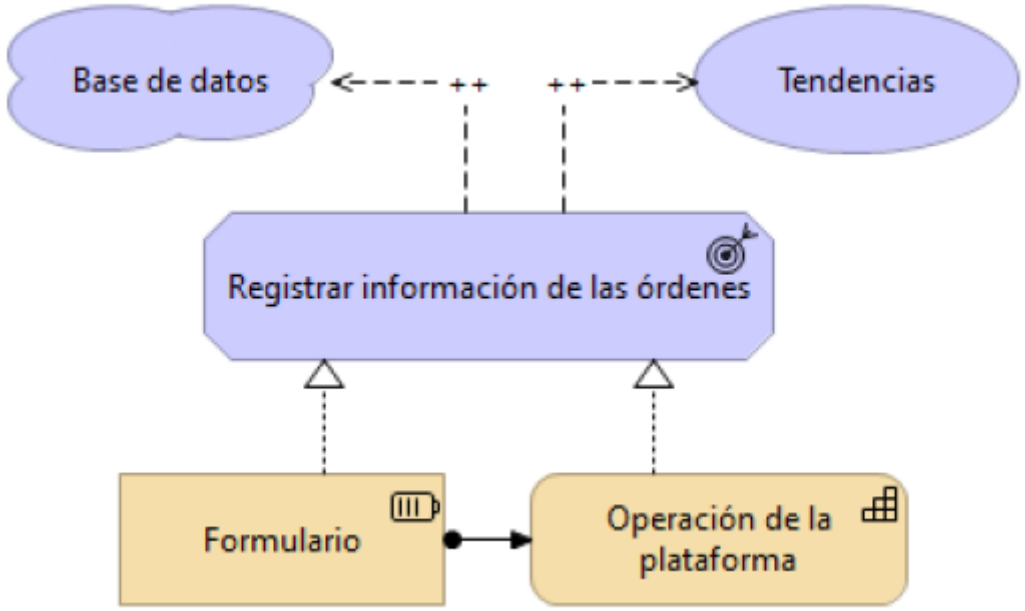



*Figura 8. Capa de Estrategia, Punto de vista de Realización de resultados. (Elaboración propia)*

### *2.2.3 Punto de vista Mapa de recurso.*

El punto de vista de mapa de recurso permite tener una visión general estructurada de los recursos de la organización describiendo las relaciones entre los recursos y las capacidades a las que están asignados.

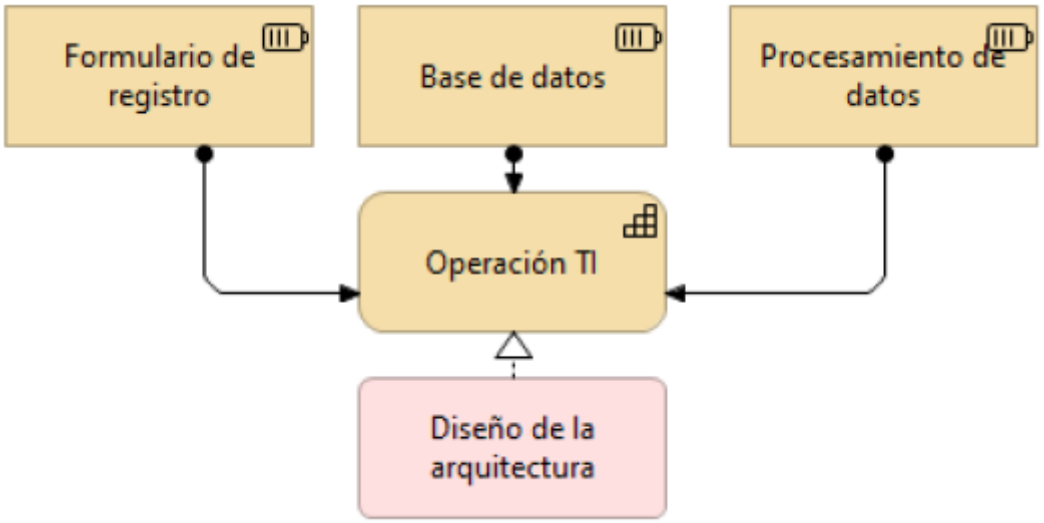



*Figura 9. Capa de Estrategia, Punto de vista de Mapa de Recurso. (Elaboración propia)*

## 2.3 Capa de Negocio

Esta capa consiste en entender los conceptos que definen la estructura estática de la organización en términos de las entidades que componen la organización y las interacciones presentadas entre estas. Entidades como actores de negocio, roles de negocio, funciones de negocio, etc, que se comportan por medio de un diseño orientado al servicio, es decir, las funciones o procesos de negocio interno y externo de la organización como las responsabilidades.

### *2.3.1 Punto de Vista de la Organización.*

El Punto de Vista de Organización se centra en la organización interna de una empresa, un departamento, una red de empresas o de otra entidad organizativa. Se utiliza para identificar las competencias, la autoridad y las responsabilidades en una organización.

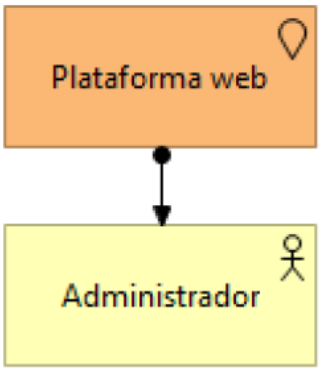



*Figura 10. Capa de Negocio, Punto de vista de la organización. (Elaboración propia)*

### *2.3.2 Punto de Vista de Cooperación de Actor*

El Punto de Vista de Cooperación de Actor se centra en las relaciones de los actores entre sí y su entorno. Se utiliza para determinar las dependencias externas y colaboraciones, y muestra la cadena de valor o la red en la que actúa el actor.

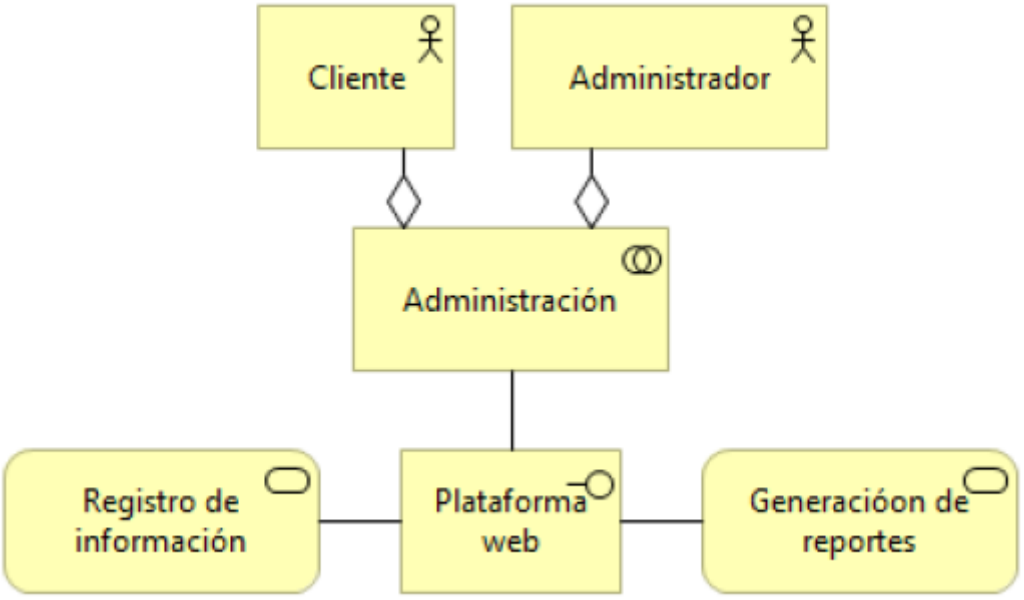



*Figura 11. Capa de Negocio, Punto de vista de cooperación del actor. (Elaboración propia)*

### *2.3.3 Punto de Vista de Función del negocio*

El Punto de Vista de la Función de Negocio muestra las principales funciones de negocio de una organización y sus relaciones en términos de los flujos de información, valor o bienes entre ellos. Adicionalmente, proporciona una visión de alto nivel de las operaciones

generales de la empresa y puede utilizarse para identificar las competencias necesarias o estructurar una organización de acuerdo con sus principales actividades.

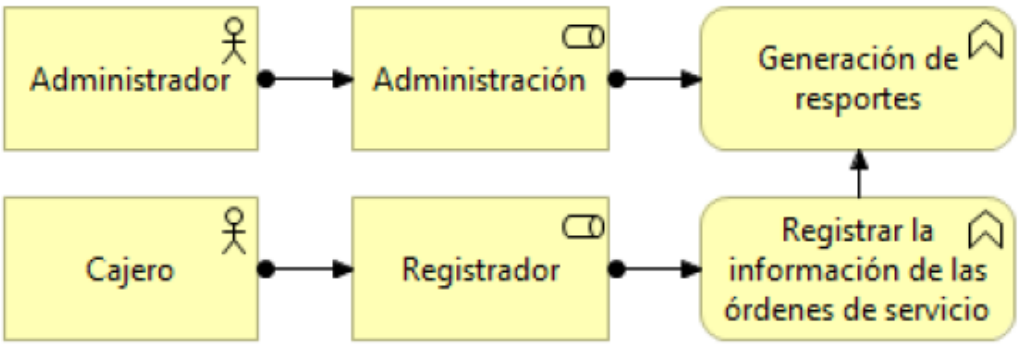



*Figura 12. Capa de Negocio, Punto de vista de Función del negocio. (Elaboración propia)*

### *2.3.4 Punto de Vista de Proceso del negocio*

El Punto de Vista de Proceso de Negocio se utiliza para mostrar la estructura y composición de alto nivel de uno o más procesos empresariales.

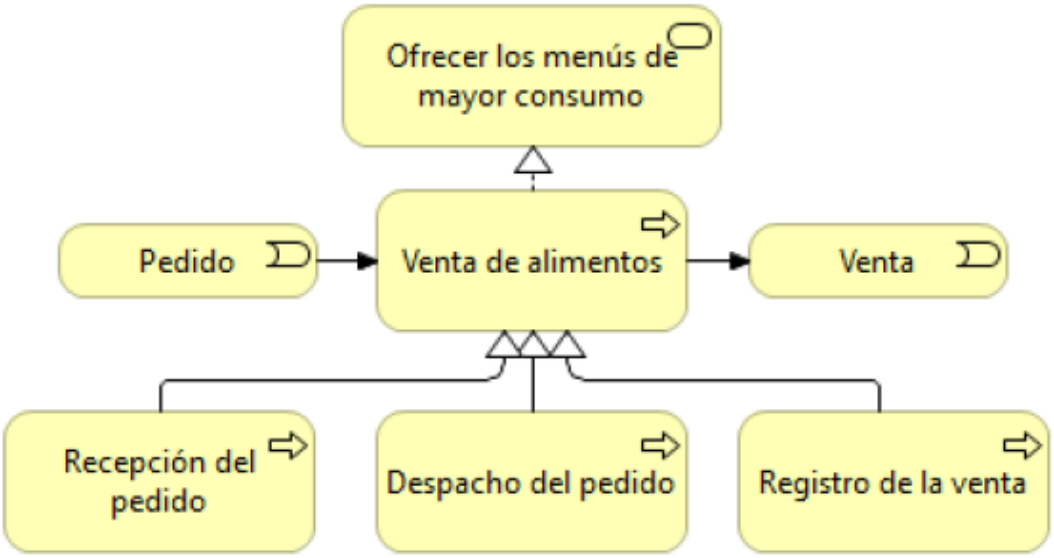



*Figura 13. Capa de Negocio, Punto de vista de Proceso de negocio. (Elaboración propia)*

### *2.3.5 Punto de Vista de Cooperación del Proceso de negocio*

El Punto de Vista de Cooperación de Proceso de Negocio se utiliza para mostrar las relaciones de uno o más procesos de negocio entre si y/o con su entorno.

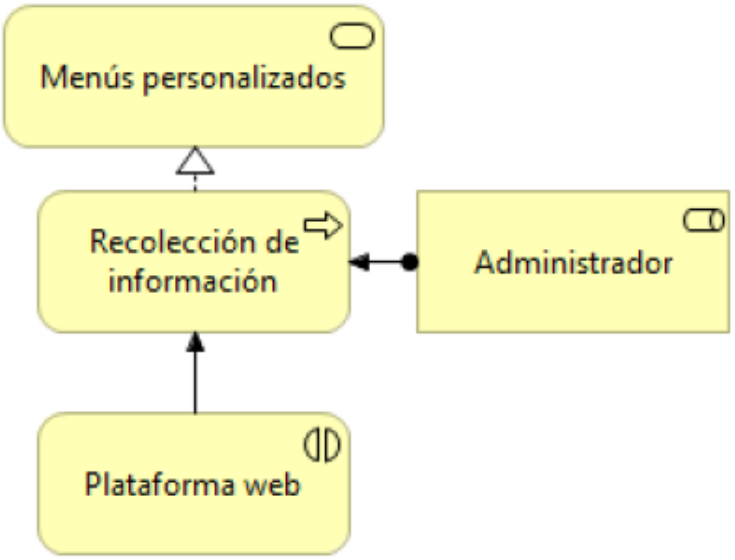



*Figura 14. Capa de Negocio, Punto de vista de Cooperación del Proceso de negocio. (Elaboración propia)*

### *2.3.6 Punto de Vista de Producto.*

El Punto de Vista de Producto representa el valor que estos productos ofrecen a los clientes u otras partes externas involucradas.

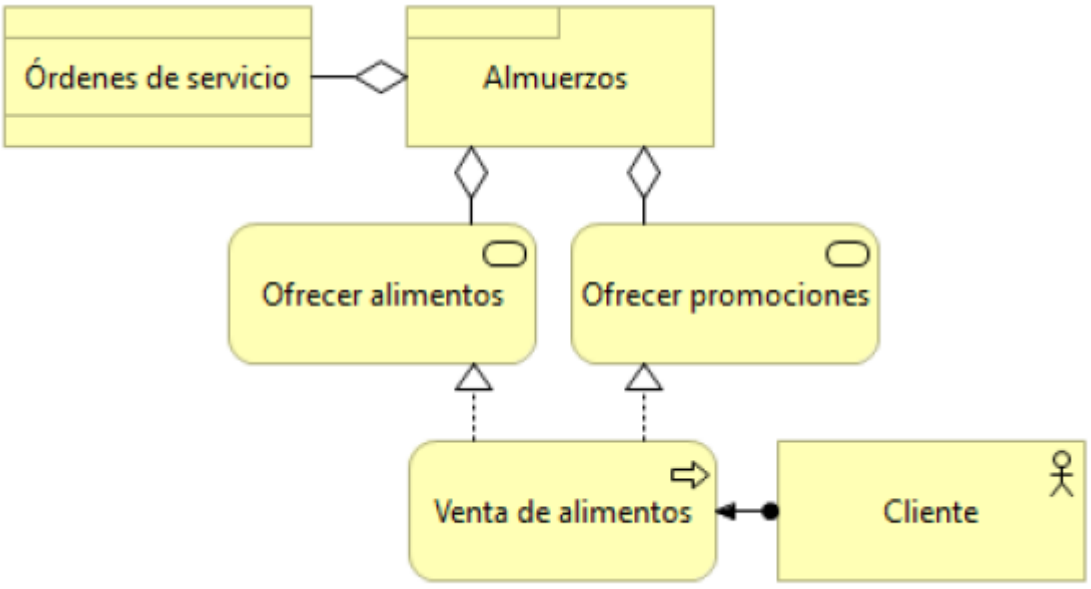



*Figura 15. Capa de Negocio, Punto de vista de Producto. (Elaboración propia)*

## 2.4 Capa de Aplicación

Esta capa retoma la mayoría de conceptos utilizados en la capa de negocio con la distinción de los diferentes usos e interacciones que se presentan. Para una adecuada apropiación de los conceptos de esta capa, se implementa el uso de cuatro diferentes

puntos de vista: comportamiento, cooperación, estructura y uso de la aplicación. Entidades como componentes y objeto de datos que se comportan por medio de una relación explícita en el modelo, es decir, el comportamiento a nivel interno y externo de la organización; el comportamiento externo dado en términos de los servicios de aplicación mientras que el comportamiento interno en funciones de aplicación que realizan estos servicios.

### *2.4.1 Punto de Vista Comportamiento de la aplicación.*

El Punto de Vista de comportamiento de la aplicación describe el comportamiento interno de una aplicación, es útil para diseñar el comportamiento principal de las aplicaciones o para identificar la superposición funcional entre las mismas.

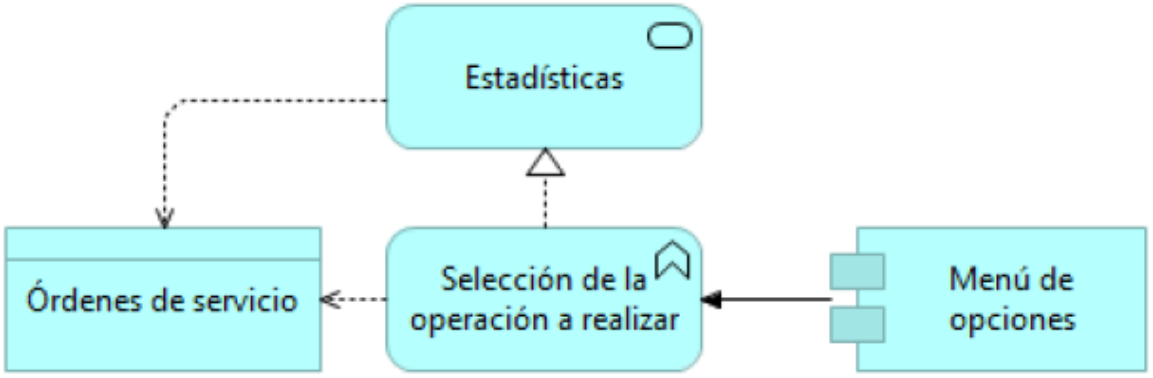



*Figura 16. Capa de Aplicación, Punto de vista de Comportamiento de la aplicación. (Elaboración propia)*

### *2.4.2 Punto de Vista Cooperación de la aplicación.*

El Punto de Vista de cooperación de la aplicación describe las relaciones entre los componentes de las aplicaciones en términos de los flujos de información entre ellos o en términos de los servicios que ofrecen y utilizan.

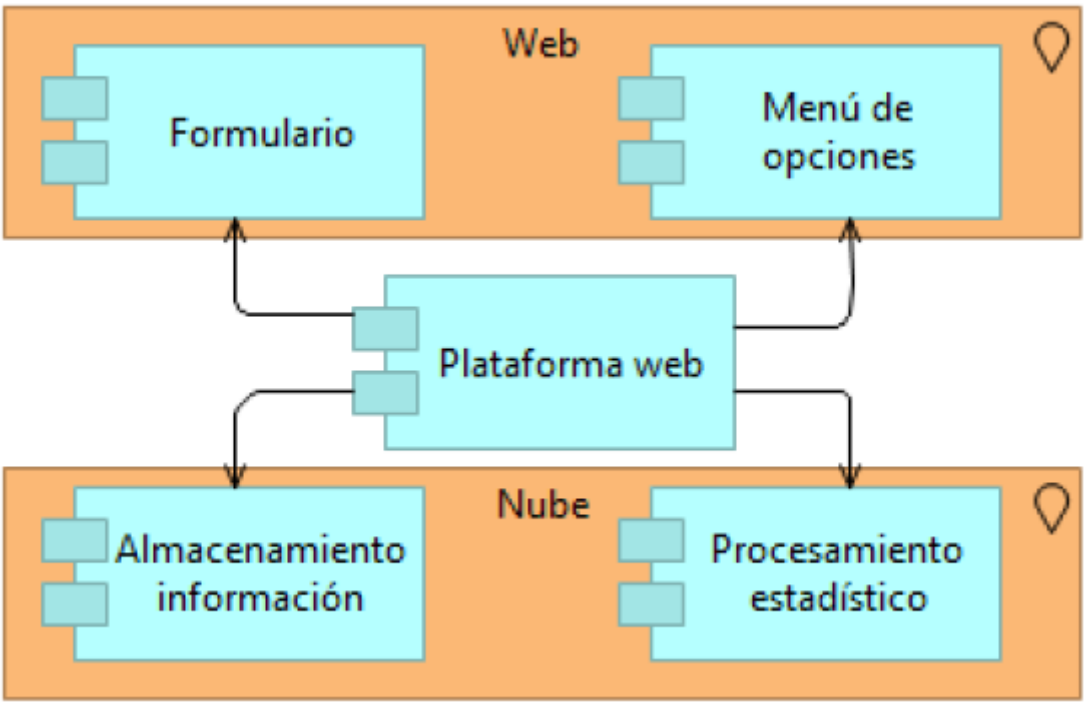



*Figura 17. Capa de Aplicación, Punto de vista de Cooperación de la aplicación. (Elaboración propia)*

### *2.4.3 Punto de Vista Estructura de la aplicación.*

El Punto de Vista de estructura de la aplicación describe la estructura principal de aplicaciones o componentes y los datos asociados.

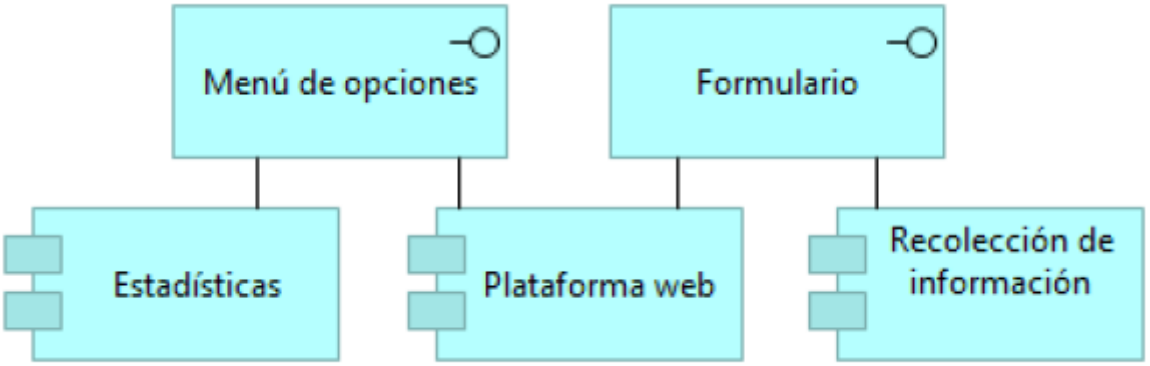



*Figura 18. Capa de Aplicación, Punto de vista de Estructura de la aplicación. (Elaboración propia)*

### *2.4.4 Punto de Vista Uso de la aplicación.*

El Punto de Vista del uso de la aplicación describe el diseño de una aplicación mediante la identificación de los servicios necesarios por los procesos de negocio y otras aplicaciones, o en el diseño de procesos de negocio mediante la descripción de los servicios que están disponibles.

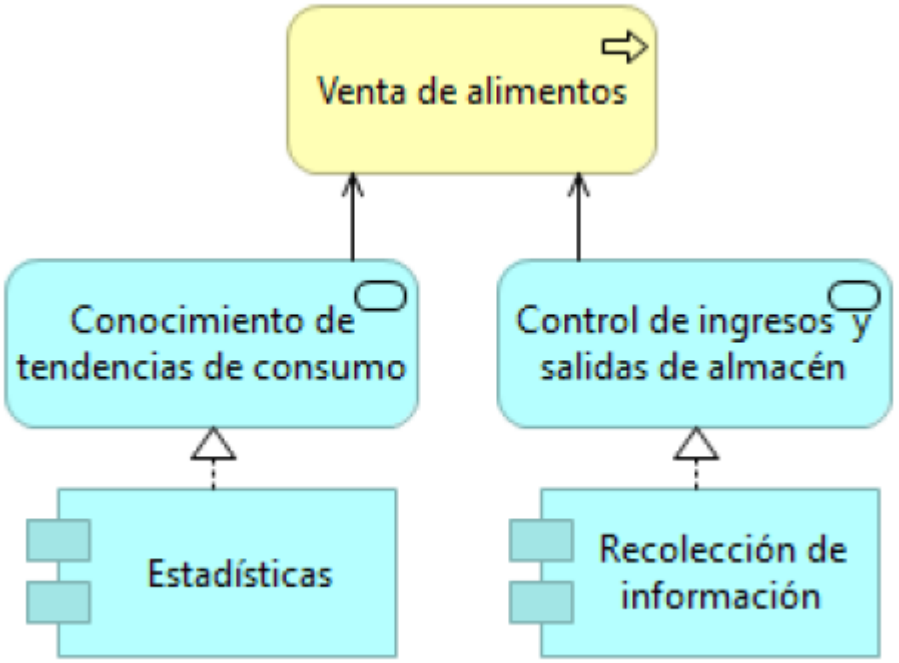



*Figura 19. Capa de Aplicación, Punto de vista de Uso de la aplicación. (Elaboración propia)*

**2.5 Capa de Tecnología**

Esta capa toma, en primer lugar, la utilización de conceptos como Nodo y Dispositivo para identificar las diferentes entidades físicas o recursos computacionales de la infraestructura del sistema. Esta capa busca establecer y conocer los diferentes elementos a nivel de software y hardware que interactúan en los procesos de negocio establecidos anteriormente, por lo tanto, es importante tener en cuenta los conceptos utilizados en la capa de negocio y en la capa de aplicación para así facilitar la identificación de los diferentes usos e interacciones que se presentan con esta capa. Para una adecuada apropiación de la conceptualización de esta capa, se implementa el uso de seis diferentes puntos de vista: infraestructura, uso de infraestructura, implementación y organización, estructura de información, realización de servicio y capas.

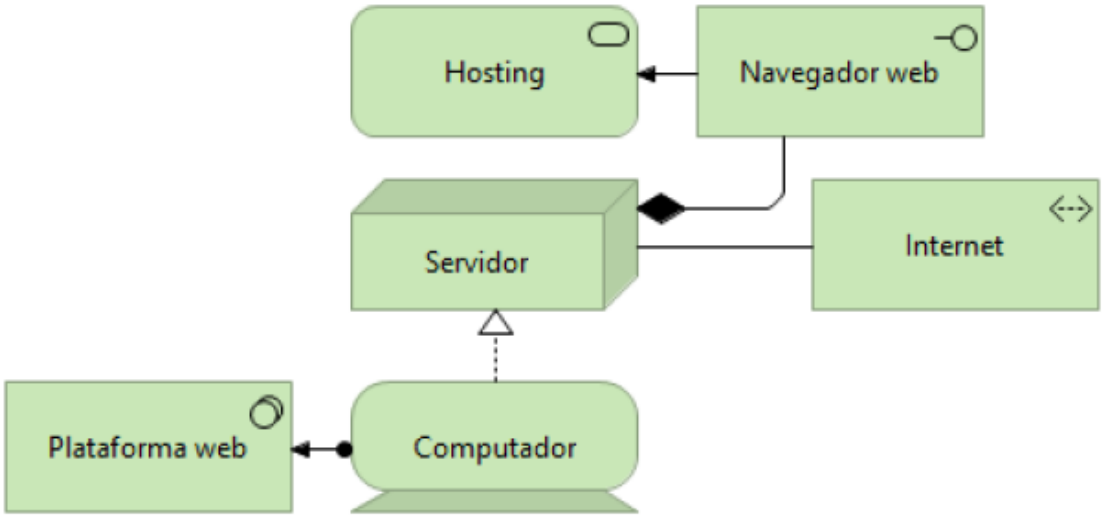



*Figura 20. Capa de Tecnología, Punto de vista de Uso de la tecnología. (Elaboración propia)*

### *2.5.1 Punto de vista de la infraestructura.*

El Punto de Vista de la Infraestructura contiene los elementos de infraestructura de software y hardware que soportan la capa de aplicación, como dispositivos físicos, redes o software del sistema.

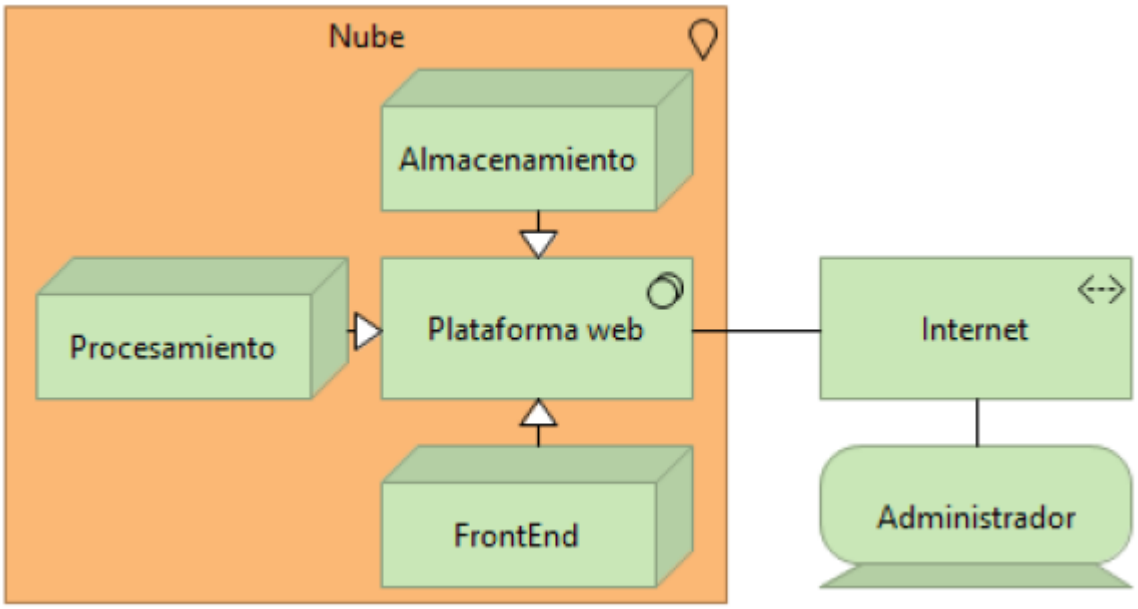



*Figura 21. Capa de Tecnología, Punto de vista de la infraestructura. (Elaboración propia)*

### *2.5.2 Punto de vista de uso de tecnología.*

El Punto de Vista del uso de la tecnología describe cómo las aplicaciones son compatibles con la infraestructura de software y hardware. Es muy útil para determinar los

requisitos de rendimiento y calidad de la infraestructura basados en las demandas de las diversas aplicaciones que la utilizan.

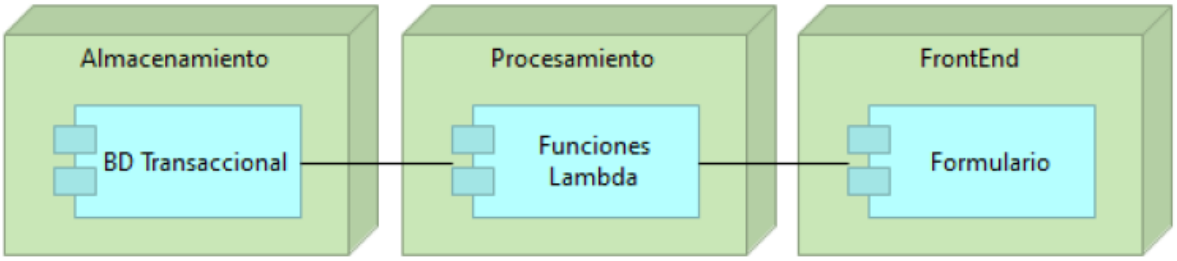



*Figura 22. Capa de Tecnología, Punto de vista de la infraestructura. (Elaboración propia)*

### *2.5.3 Punto de vista de implementación y organización.*

El Punto de Vista de implementación y Organización describe cómo se despliegan una o más aplicaciones en la infraestructura. Este punto de vista comprende la asignación de aplicaciones y componentes lógicos a artefactos físicos.

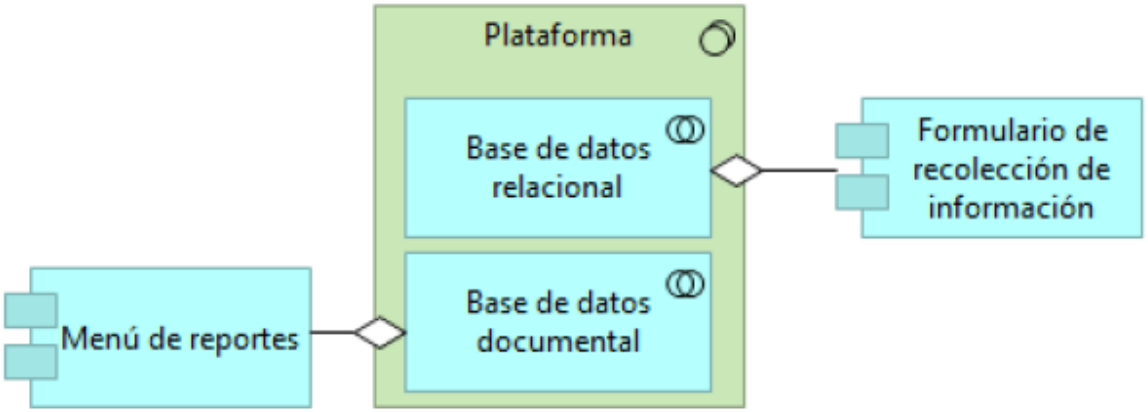



*Figura 23. Capa de Tecnología, Punto de vista de implementación y organización.. (Elaboración propia)*

### *2.5.4 Punto de vista de estructura de información.*

El Punto de Vista de Estructura de la Información describe la estructura de la información usada en la empresa o en un proceso específico de negocio o aplicación, en términos de tipos de datos o estructuras de clases.

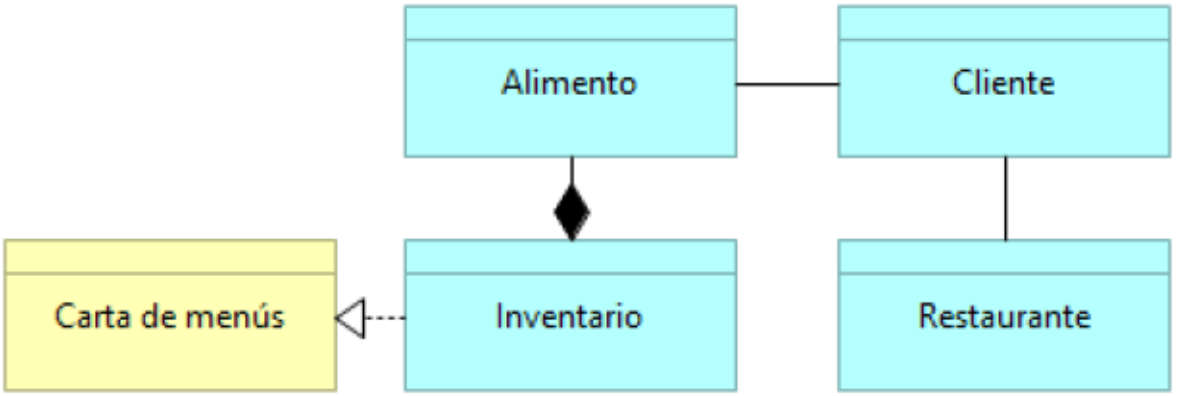



*Figura 24. Capa de Tecnología, Punto de vista de estructura de información. (Elaboración propia)*

### *2.5.5 Punto de vista de realización de servicio.*

El Punto de Vista de Realización de Servicio describe cómo uno o más servicios de negocio son realizados por un proceso fundamental o, algunas veces, por componentes de aplicación. Esto forma el puente entre productos y las vistas de los procesos de negocio.

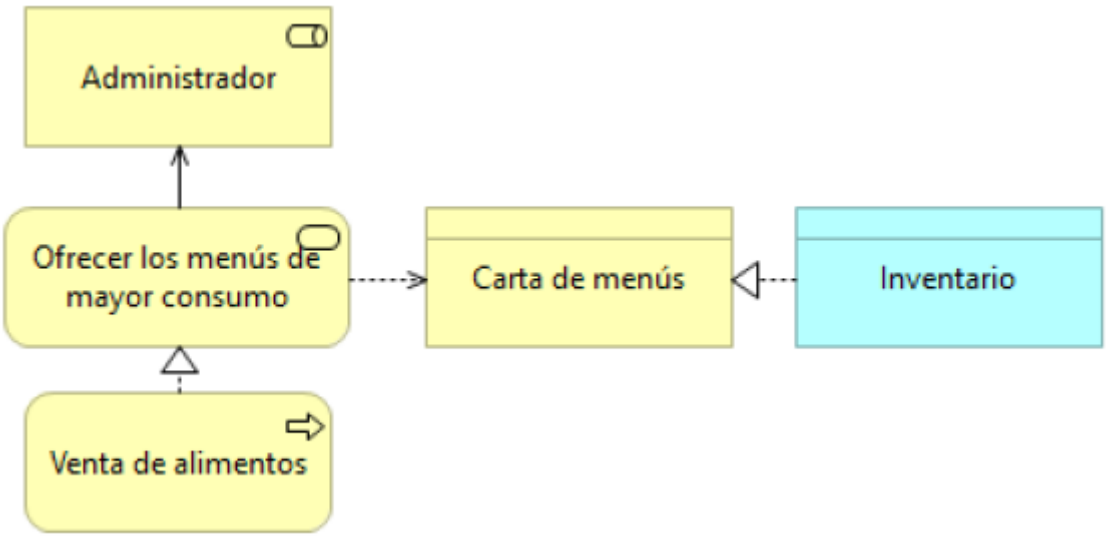



*Figura 25. Capa de Tecnología, Punto de vista de realización del servicio. (Elaboración propia)*

### *2.5.6 Punto de vista de las capas.*

El Punto de Vista de capas describe varios aspectos de una arquitectura empresarial en un diagrama como resultado del uso de la relación de agrupación para una partición natural de todo el conjunto de objetos y relaciones que pertenecen al modelo.




*Figura 26. Capa de Tecnología, Punto de vista de Capas. (Elaboración propia)*

## 2.6 Capa de Implementación y despliegue

Esta capa describe la implementación de modelos y características de migración para ejecutar un cambio en la arquitectura del negocio. Cada uno de los puntos de vista que contiene esta capa presenta una perspectiva para modelar la gestión del cambio de arquitectura, modelar la transición de una arquitectura existente a una arquitectura de destino y definir las relaciones entre los programas y proyectos y las partes de la arquitectura que implementan.

### *2.6.1 Punto de vista de Proyecto*

El punto de vista de proyecto se utiliza principalmente para modelar la gestión del cambio de la arquitectura permitiendo al dise˜nador o analista plasmar mediante un diagrama el proceso de interacci´on de los diferentes elementos




*Figura 27. Capa de Implementación y despliegue, Punto de vista de Proyecto. (Elaboración propia)*

### *2.6.2 Punto de vista de Migración*

El Punto de Vista de Migración implica modelos y conceptos que pueden usarse para especificar la transición de una arquitectura existente a una arquitectura deseada.

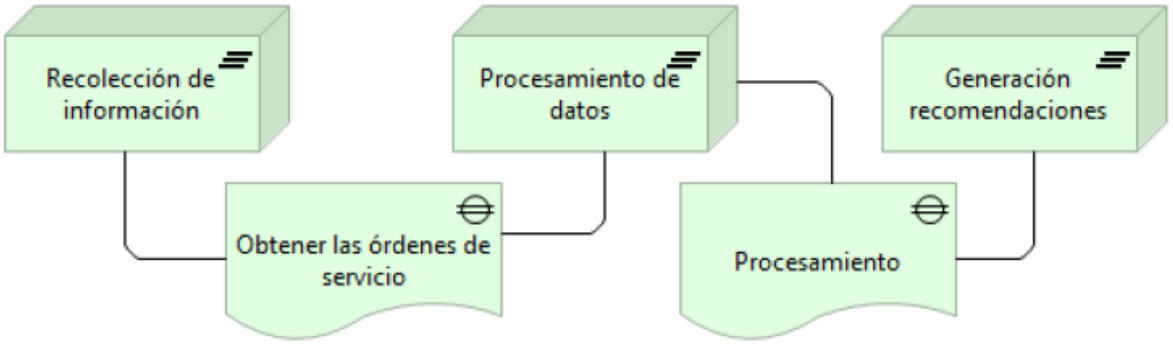



*Figura 28. Capa de Implementación y despliegue, Punto de vista de Migración. (Elaboración propia)*

### *2.6.3 Punto de vista de Implementación/Migración*

El Punto de Vista de Migración e Implementación se utiliza para relacionar programas y proyectos con las partes de la arquitectura que implementan

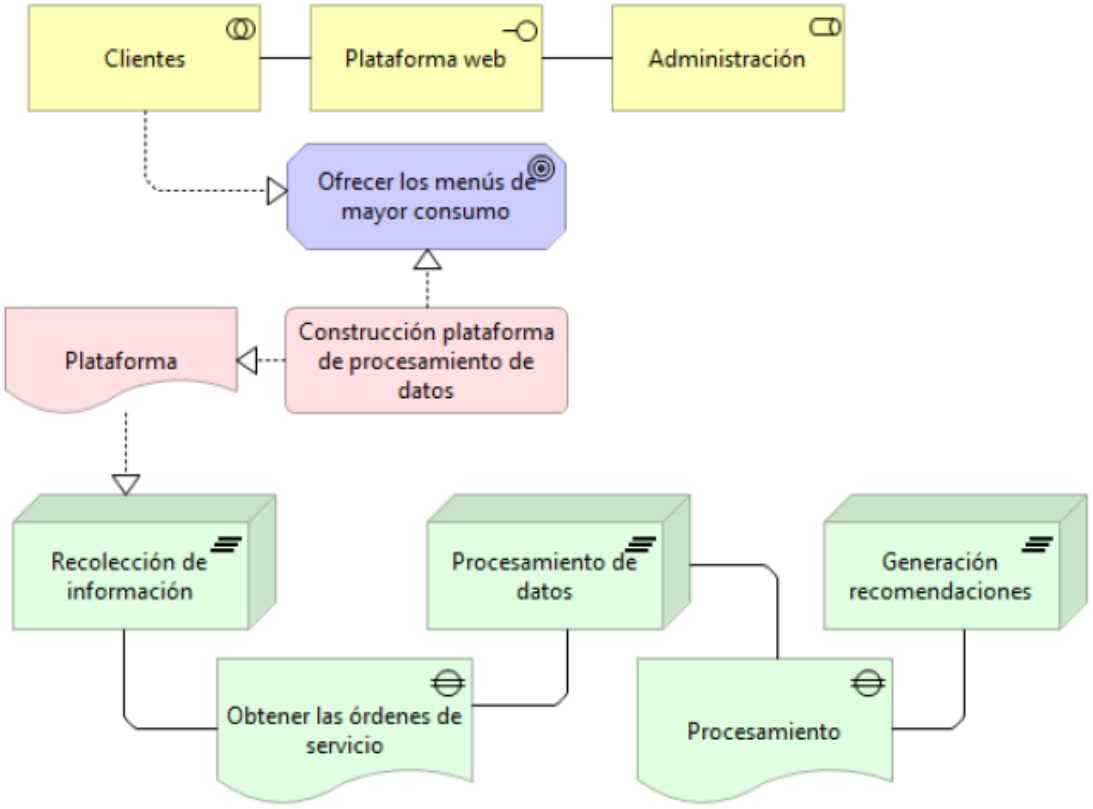



*Figura 29. Capa de Implementación y despliegue, Punto de vista de Implementación / Migración. (Elaboración propia)*

## 2.7 Capa Física

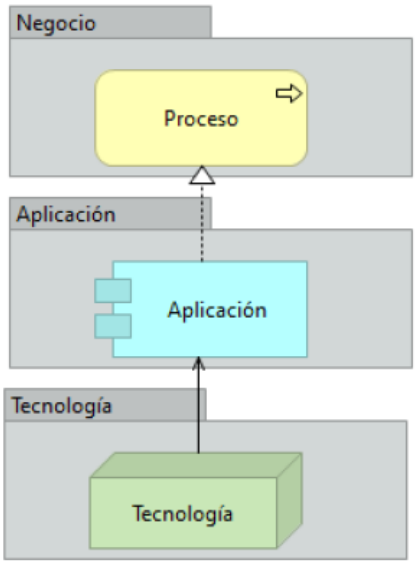



*Figura 30. Capa Física. (Elaboración propia)*

# CAPÍTULO 3. DISEÑO DEL PROTOTIPO

## 3.1 Modelo Lógica de Negocio

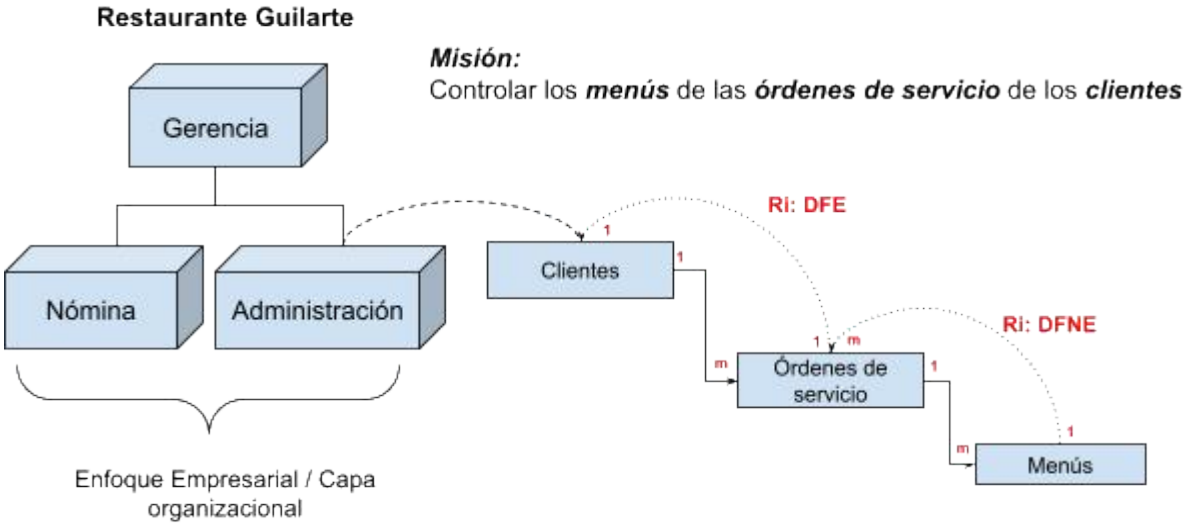



*Figura 31. Modelo de Lógica de Negocio (Elaboración propia)*

En la figura anterior se evidencia el modelo de la lógica de negocio que se plantea para el Restaurante Guilarte, consiste en una arquitectura monomisional extendida, consta de un área administrativa cuya única misión es conocer la información de los platillos vendidos por cada una de las órdenes de servicio para cada uno de los clientes.

En el modelo se puede observar que la relación directa entre las clases objetivo (clientes, órdenes y menús) se describe de la siguiente manera: Para cada uno de los clientes hay muchas órdenes de servicio, y cada una de las órdenes de servicio tiene muchos menús dentro de ella.

La relación inversa indica que cada uno de los menús puede estar en muchas órdenes *(Dependencia Funcional No Exclusiva - DFNE)*, pero, cada una de esas órdenes, le pertenecen exclusivamente a un solo cliente *(Dependencia Funcional Exclusiva)*.

Este modelo permite generar consultas relacionadas con top de menús consumidos, top de menús que más valor representan, clientes que más aportan, los menús que más consume determinado cliente, y otra serie de reportes de las que se hablará más adelante.

## 3.2 Modelo Base de datos sugerido

### *3.2.1 Base de datos MySQL*

Basados en el modelo de la lógica de negocio *(MLN)* del apartado anterior, específicamente en la *Figura 31,* la base de datos que permite construirla, consta de las tablas descritas con sus atributos en el modelo seudomatemático presentado a continuación en la *Figura 32*:

***Clientes:***
- *{* ***id_cliente (PK)***, nombre_cliente, telefono_cliente, direccion_cliente, correo_cliente *}*

***Ordenes_servicio:***
- { ***id_orden (PK)***, ***id_cliente (FKP)***, valor_pagado, fecha_orden, *id_tipo_servicio (FKD)*, *id_mesero (FKD)*, *id_mesa (FKD)* }

***Tipos_servicio:***
- { ***id_tipo_servicio (PKE)***, nombre_tipo_servicio }

***Meseros:***
- { ***id_mesero (PKE)***, nombre_mesero, identificacion_mesero }

***Mesas:***
- { ***id_mesa (PKE)***, numero_sillas }

***OrdenesServicioMenus:***
- { ***(id_orden (FKP), id_menu (FKP)) PK***, cantidad_menu_en_orden }

***Menus:***
- { ***id_menu (PK)***, nombre_menu, valor_menu, *id_disponibilidad (FKD)*, *id_tipo_menu (FKD)* }

***Tipos_menu:***
- { ***id_tipo_menu (PKE)***, nombre_tipo_menu }

***Tipos_Disponibilidad:***
- { ***id_disponibilidad (PKE)***, nombre_disponibilidad }

*Figura 32. Modelo Seudomatemático (Elaboración propia)*

Una vez se ha definido el modelo seudomatemático donde se evidencia la solución de las dependencias, se construye el modelo relacional evidenciado a continuación en la *Figura 33:*

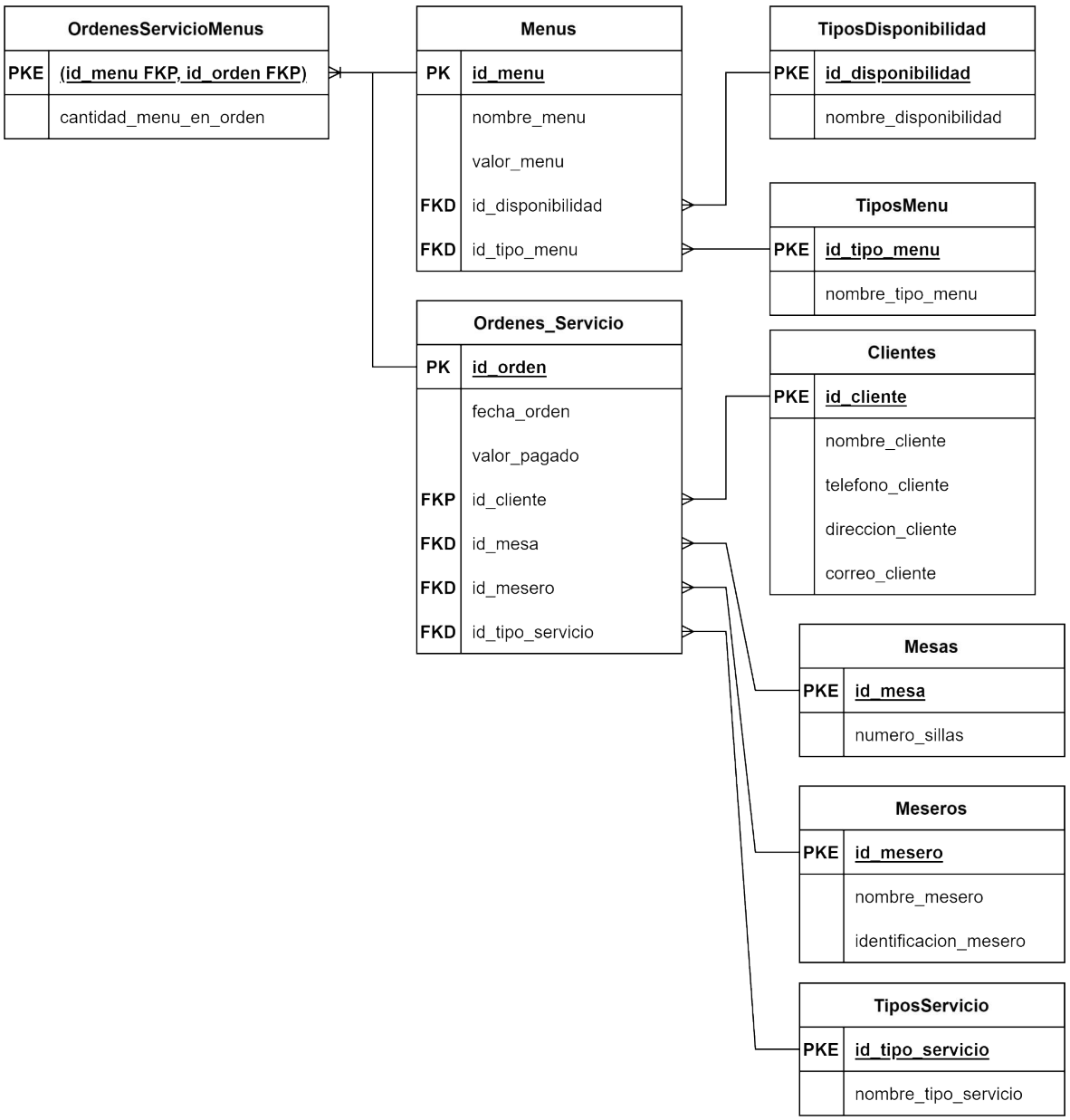



*Figura 33. Modelo Relacional (Elaboración propia)*

Esta base de datos permitirá almacenar todas las transacciones relacionadas con las órdenes de servicio, a través de un formulario que se describe más adelante en la sección *Interfaz de Usuario.*

### *3.2.2 Base de datos NoSQL*

Teniendo en cuenta la finalidad del proyecto en la generación de reportes, es necesario guardar el registro de la información sobre la que se ha consultado, esta información puede ser tratada con facilidad en el futuro para exportar a documentos gerenciales, por tal razón, se ha diseñado la siguiente base de datos NoSQL descrita en la *Figura 34:*

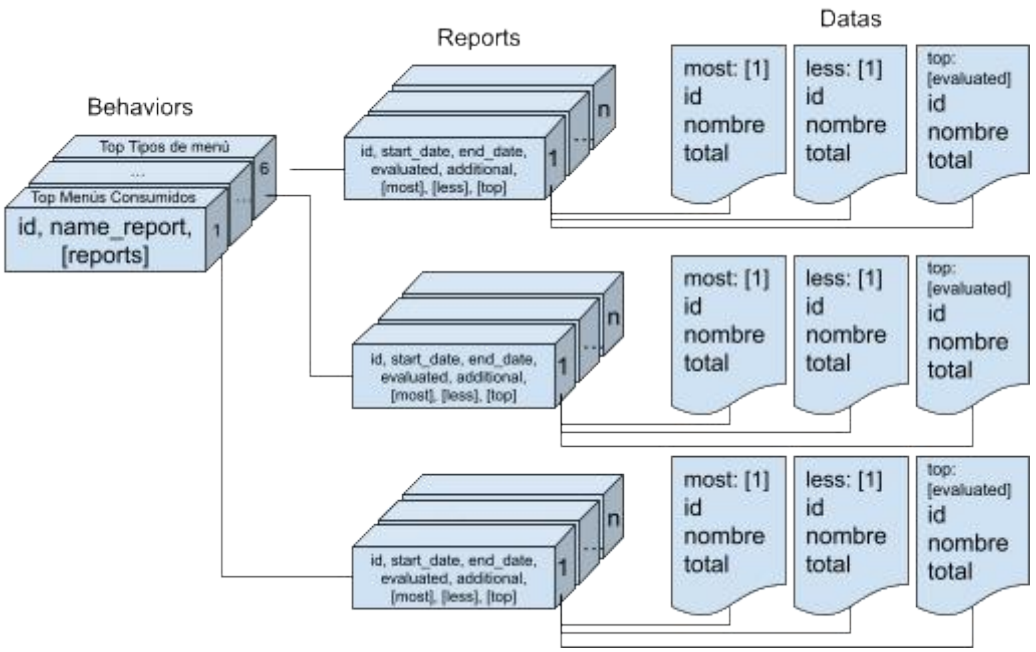



*Figura 34. Diagrama Base de datos documental (Elaboración propia)*

Esta base de datos se encuentra construída con 3 colecciones básicas:

1. ***Behaviors:*** Hace referencia a los tipos de comportamientos que se almacenarán.
2. ***Reports:*** Hace referencia al reporte específico de determinado análisis solicitado.
3. ***Datas:*** Hace referencia a las llaves nombre, total, cuyo contenido depende igualmente del análisis solicitado.

Como se puede observar, cada comportamiento tiene una llave *reports* que es un arreglo de identificadores, a su vez, cada uno de los reportes tiene unas llaves llamadas *most, less, top* que son un arreglo de identificadores de *datas.*

### 3.3 Arquitectura de software

La *Figura 35* representa la arquitectura de diseño del prototipo, en ella se puede observar que se usan 2 proveedores de infraestructura *(Retool, Railway)*, cada una de ellas alojando los

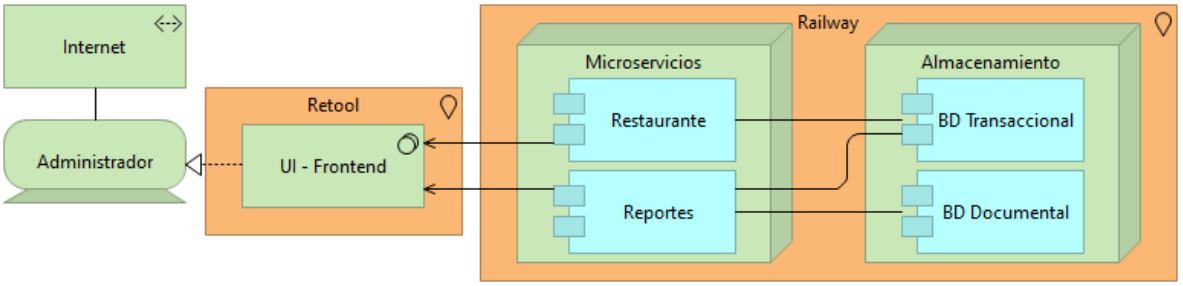



*Figura 35. Arquitectura del prototipo (Elaboración propia)*

***Retool:*** De acuerdo con (G2, 2022), retool es una plataforma *low-code* que permite la construcción de herramientas internas de una manera más fácil y rápida, ya que entre sus productos proveé:

- Componentes *(botones, tablas, formularios, gráficas)*, plantillas.
- Integración con diferentes servicios, incluyendo bases de datos a través de *REST, GraphQL, APIs.*

***Railway:*** Es una plataforma en la nube que permite el aprovisionamiento de infraestructura, permite el despliegue de:

- Proyectos alojados en repositorios de github.
- Bases de datos PostgreSQL, Redis, MongoDB, MySQL.

## CAPÍTULO 4. IMPLEMENTACIÓN DEL PROTOTIPO

### 4.1 Base de datos

Gracias al aprovisionamiento de *Railway,* se crearon 2 proyectos en esta plataforma para alojar las bases de datos, tanto la transaccional como la documental, el cliente de base de datos usado para interactuar con la base de datos es *DBeaver* que es una herramienta multiplataforma para administración de bases de datos, el script de creación se presenta en la *Figura 36,* a continuación:

```
USE railway;
-- Tipos de servicio
CREATE TABLE IF NOT EXISTS tipos_servicio(
		id_tipo_servicio INT(2) PRIMARY KEY AUTO_INCREMENT,
		nombre_tipo_servicio VARCHAR(12)
);
-- Meseros
CREATE TABLE IF NOT EXISTS meseros(
		id_mesero INT(2) PRIMARY KEY AUTO_INCREMENT,
		nombre_mesero VARCHAR(32),
		identificacion_mesero INT(10)
);
-- Mesas
CREATE TABLE IF NOT EXISTS mesas(
		id_mesa INT(2) PRIMARY KEY AUTO_INCREMENT,
		numero_sillas INT(2)
);
-- Clientes
CREATE TABLE IF NOT EXISTS clientes(
		id_cliente INT(3) PRIMARY KEY AUTO_INCREMENT,
		nombre_cliente VARCHAR(32),
		telefono_cliente BIGINT(10),
		direccion_cliente VARCHAR(32),
		correo_cliente VARCHAR(32)
);
-- Tipos de menú
CREATE TABLE IF NOT EXISTS tipos_menu(
		id_tipo_menu INT(2) PRIMARY KEY AUTO_INCREMENT,
		nombre_tipo_menu VARCHAR(32)
);

-- Disponibilidad
CREATE TABLE IF NOT EXISTS tipos_disponibilidad(
		id_disponibilidad INT(2) PRIMARY KEY AUTO_INCREMENT,
		nombre_disponibilidad VARCHAR(32)
);
-- Órdenes de servicio
CREATE TABLE IF NOT EXISTS ordenes_servicio(
		id_orden INT(5) PRIMARY KEY AUTO_INCREMENT,
		fecha_orden TIMESTAMP DEFAULT NOW(),
		valor_pagado INT(12),
		id_cliente INT(3),
		id_mesa INT(2),
		id_mesero INT(2),
		id_tipo_servicio INT(2),
		FOREIGN KEY (id_cliente) REFERENCES clientes(id_cliente),
		FOREIGN KEY (id_mesa) REFERENCES mesas(id_mesa),
		FOREIGN KEY (id_mesero) REFERENCES meseros(id_mesero),
		FOREIGN KEY (id_tipo_servicio) REFERENCES tipos_servicio(id_tipo_servicio)
);
-- Menús
CREATE TABLE IF NOT EXISTS menus(
		id_menu INT(5) PRIMARY KEY AUTO_INCREMENT,
		nombre_menu VARCHAR(64),
		valor_menu INT(9),
		id_disponibilidad INT(2),
		id_tipo_menu INT(2),
		FOREIGN KEY (id_disponibilidad) REFERENCES tipos_disponibilidad(id_disponibilidad),
		FOREIGN KEY (id_tipo_menu) REFERENCES tipos_menu(id_tipo_menu)
);
-- Menús de las órdenes de servicio
CREATE TABLE IF NOT EXISTS ordenes_servicio_menus(
		id_orden INT(5),
		id_menu INT(5),
		cantidad_menu_en_orden INT(3),
		PRIMARY KEY (id_orden, id_menu),
		FOREIGN KEY (id_orden) REFERENCES ordenes_servicio(id_orden),
		FOREIGN KEY (id_menu) REFERENCES menus(id_menu)
);
```

*Figura 36. Script creación tablas base de datos (Elaboración propia)*

Mediante la misma herramienta *DBeaver* se puede observar el modelo relacional, visto en la *Figura 37*:

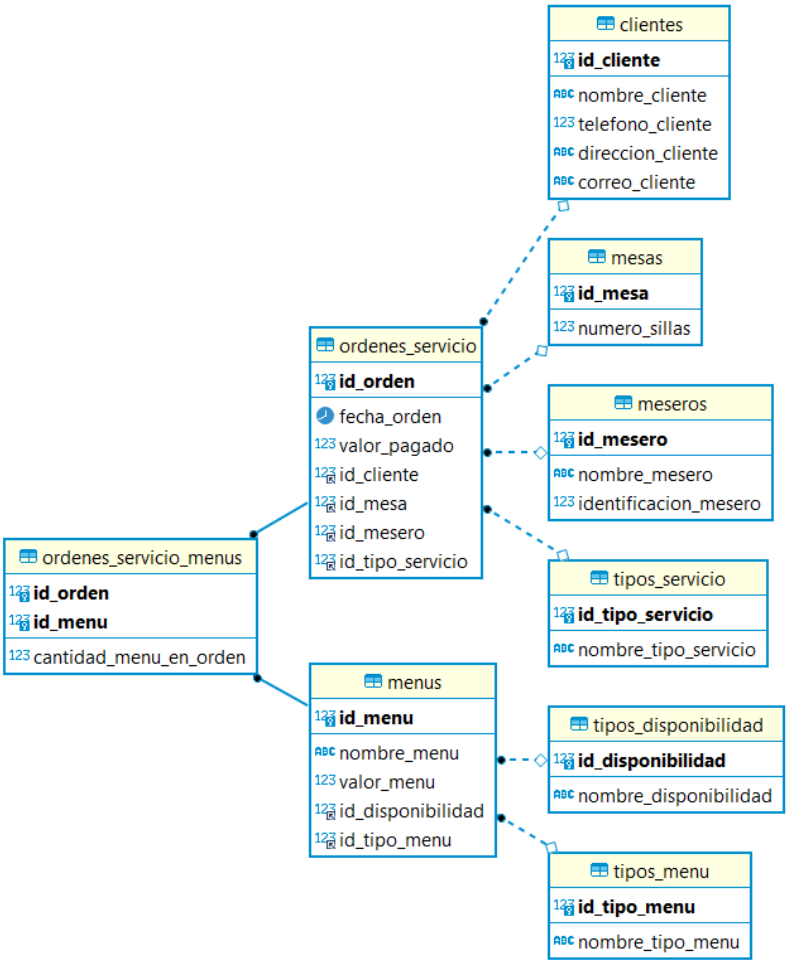



*Figura 37. Modelo Relacional Base de datos SQL (DBeaver)*

En cuanto a la creación de las colecciones para la base de datos documental se usó la interfaz que proveé *Railway* para ello, interfaz vista en la *Figura 38:*

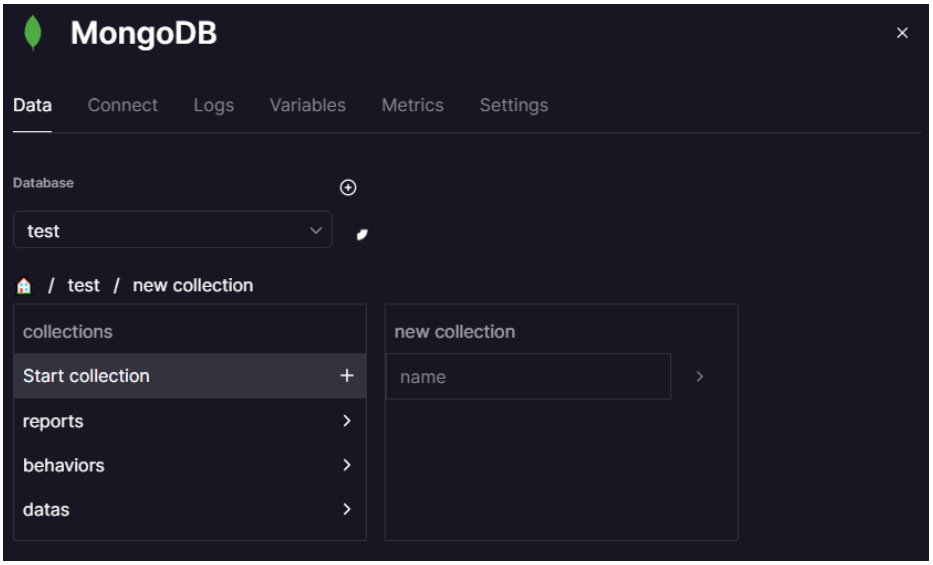



*Figura 38. Creación colecciones Base de datos NoSQL (Railway)*

### 4.2 Interfaz de usuario

La interfaz de usuario se construyó aprovechando la facilidad con la que se pueden desarrollar aplicaciones en *Retool,* su construcción consiste en el *Drag&Drop* de componentes suministrados, teniendo en cuenta que para el proyecto se requirieron 2 secciones *(Órdenes de servicio, Reportería)*, se utilizaron los siguientes componentes para cada una de las secciones:

***Órdenes de servicio:*** Para este módulo se hizo uso de:

- *Formulario:* Elemento principal de la sección, ya que toda la información diligenciada corresponde a la información que se almacenará en la base de datos.
- *Botones:* Componentes para ejecutar las acciones de añadir menús a la órden de servicio.
- *Listas desplegables:* Este componente se usó para seleccionar información dentro del formulario, su contenido viene del microservicio *Restaurante*, del cuál se hablará más adelante. Las listas se usaron para obtener:
    - Meseros
    - Mesas
    - Tipos de servicio
    - Menús
- *Campos de texto / número:* Permite introducir manualmente valores para la siguiente información del formulario:
    - Nombre del cliente
    - Teléfono del cliente
    - Selector de cantidad de un determinado menú
- *Tablas:* Se usa el componente tabla para listar los menús escogidos, junto con su valor unitario, y el valor total en función de la cantidad seleccionada.

***Reportería:*** Para este módulo se usaron los siguientes componentes:

- *Checkbox:* Este componente permite seleccionar exclusivamente una de las opciones, sus valores hacen referencia al tipo de reporte que se desea.
- *Formulario:* Para obtener un reporte, es necesario enviar los parámetros, por ello, un formulario es la mejor opción para recolectarlos.
- *Entrada de fecha:* Cada uno de los reportes requiere de un período de tiempo, así que la entrada tipo rango de fecha es la adecuada para ello.
- *Entrada de texto:* Uno de los reportes requiere la entrada del teléfono del cliente, por lo que una entrada de este tipo era necesaria, sin embargo, esta entrada se encuentra oculta para todos los reportes que no la necesiten.
- *Botón:* Este componente se usa para ejecutar la petición del reporte.
- *Gráfica de barras:* En esta gráfica se representan los datos de cada uno de los reportes, en orden descendente, teniendo en cuenta que los reportes son tops.

El resultado de la interfaz de usuario puede observarse en la sección *Síntesis del modelo propuesto*

### 4.3 Microservicio Restaurante

Para soportar la operación de negocio, es necesario obtener la información de los meseros, mesas, tipos de servicio y menús del restaurante, tal como se mencionó en el modelo de la base de datos sugerida y campos que hacen parte del formulario en la interfaz de usuario, adicionalmente, es necesario almacenar la información de las órdenes de servicio, y guardar a los clientes cuando son nuevos. Para la implementación de este microservicio y como se mencionó en el apartado de arquitectura, se desplegó en *Railway* un proyecto alojado en un repositorio de github, desarrollado en *NodeJS,* con la estructura de carpetas evidenciada en la *Figura 39*:

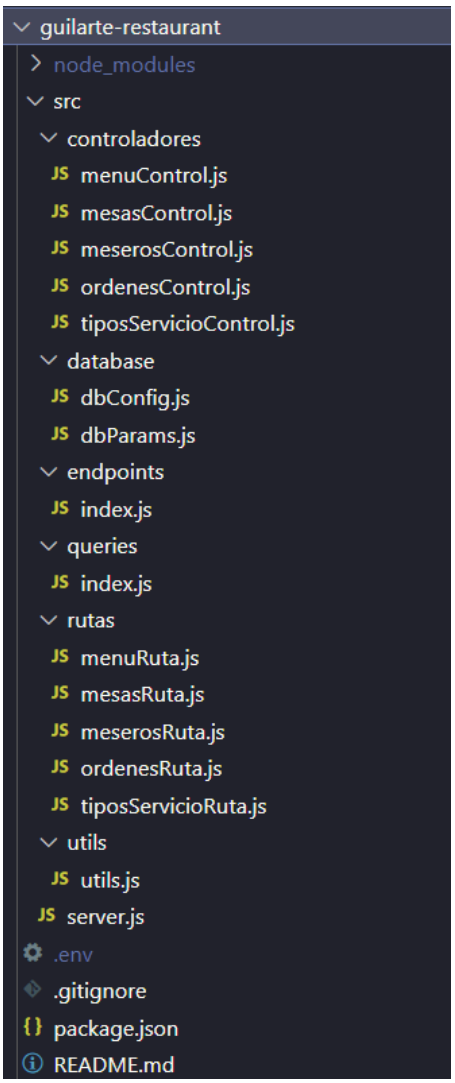



*Figura 39. Estructura de carpetas microservicio Restaurante (Elaboración propia)*

Este proyecto cuenta con las siguientes dependencias:

- *express:* Es un entorno de trabajo *NodeJS* para el desarrollo de aplicaciones web, permite la creación de un servidor.
- *mysql2:* Dependencia que permite la conexión e interacción con bases de datos SQL, a través de sus funciones permite realizar queries con la misma sintaxis *SQL.*
- *nodemon:* Esta dependencia solo se usa para desarrollo, es decir, para funcionamiento local, permite que al momento de hacer cualquier cambio y guardar el proyecto, se recompile de manera automática reiniciando el servidor que lo soporta.

A continuación se profundiza un poco sobre las carpetas de la *Figura 39:*

- ***database:*** Consta de los siguientes archivos:
    - *dbConfig.js:* Archivo de conexión de la base de datos.

```
const mysql = require("mysql2")
const { promisify } = require("util")
const { DB_PARAMS } = require("./dbParams")

const pool = mysql.createPool(DB_PARAMS)

pool.getConnection((err, connection) => {
    if(err) console.error(`errorDB: ${JSON.stringify(err)}`)
    if(connection) {
        connection.release()
        console.log("DB Connected")
        return
    }
})

pool.query = promisify(pool.query)

module.exports = pool
```

*Figura 40. Archivo de configuración conexión base de datos (Elaboración propia)*

    - *dbParams.js:* Archivo con los valores de conexión a la base de datos y nombres de las respectivas tablas.

- ***controladores:*** Se encuentran las funciones que el servidor ejecutará cuando se llame a una de las rutas.

```
const pool = require("../database/dbConfig")
const { QUERIES } = require("../queries")

module.exports = {
    async LIST (_, res) {
        try {
            const menus = await pool.query(QUERIES.GET_MENU)
            return res.status(200).json({
                menus,
            })
        } catch (err) {
            return res.status(400).json({
                mensaje: "Hubo un error con la petición",
                error: JSON.stringify(err),
            })
        }
    },
}
```

*Figura 41. Ejemplo controlador de menú (Elaboración propia)*

- ***rutas:*** Hace referencia a las petición *HTTP* que debe ejecutarse en determinado *endpoint* y la función del controlador que debe ejecutar.

```
const router = require("express").Router()
const menuController = require("../controladores/menuControl")

const { ENDPOINTS } = require("../endpoints")

router.route(ENDPOINTS.BASE).get(menuController.LIST),

module.exports = router
```

*Figura 42. Ejemplo rutas para el menú (Elaboración propia)*

- ***endpoints:*** Es el archivo donde se encuentran las variables para cada uno de los endpoints.

```
module.exports = {
    ENDPOINTS: {
        BASE: "/",
        MENU: "/menu",
        MESAS: "/mesas",
        MESEROS: "/meseros",
        TIPOS_SERVICIO: "/tipos",
        ORDENES: "/ordenes",
    },
}
```

*Figura 43. Archivo de configuración de endpoints (Elaboración propia)*

- ***queries:*** Es el archivo donde se encuentran todas las queries a ejecutarse, dependiendo de cual sea la llamada dentro del archivo de controladores.

```
const { TABLE_NAMES } = require("../database/dbParams")
const { MENU, MESAS, MESEROS, TIPOS_SERVICIO, CLIENTES } = TABLE_NAMES

module.exports = {
    QUERIES: {
        GET_MENU: `SELECT * FROM ${MENU} WHERE id_disponibilidad in (1)`,
        GET_MESAS: `SELECT * FROM ${MESAS}`,
        GET_MESEROS: `SELECT * FROM ${MESEROS}`,
        GET_TIPOS_SERVICIO: `SELECT * FROM ${TIPOS_SERVICIO}`,
        SELECT_CLIENTES: (tel) => `SELECT * FROM ${CLIENTES} WHERE telefono_cliente = ${tel} LIMIT 1`,
        INSERT_CLIENTE: (name, tel) =>
            `INSERT INTO clientes (nombre_cliente, telefono_cliente)
                VALUES ('${name}', ${tel})`,
        INSERT_ORDEN: (total, id_cliente, id_mesa, id_mesero, id_tipo_servicio) =>
            `INSERT INTO ordenes_servicio (valor_pagado, id_cliente, id_mesa, id_mesero, id_tipo_servicio)
                VALUES (${total}, ${id_cliente}, ${id_mesa}, ${id_mesero}, ${id_tipo_servicio})`,
        INSERT_ORDEN_MENUS: (id_orden, id_menu, cantidad) =>
            `INSERT INTO ordenes_servicio_menus (id_orden, id_menu, cantidad_menu_en_orden)
                VALUES (${id_orden}, ${id_menu}, ${cantidad})`,
    },
}
```

*Figura 44. Archivo de consultas a la base de datos (Elaboración propia)*

- ***utils:*** Se encuentran archivos referentes a utilidades, para este caso, función que transforma en minúsculas todos los caracteres de una cadena de palabras y

capitaliza la primera letra de cada una para devolverla como si se tratara de un nombre propio, se utiliza para normalizar la escritura de los nombres de los clientes.

El archivo más importante en este microservicio es el *server,js* que se encarga de crear el servidor y conectar las rutas.

```
require("dotenv").config()
const express = require("express")
const { ENDPOINTS } = require("./endpoints")
const menuRutas = require("./rutas/menuRuta")
const mesasRutas = require("./rutas/mesasRuta")
const meserosRutas = require("./rutas/meserosRuta")
const tiposServicioRutas = require("./rutas/tiposServicioRuta")

const ordenesRutas = require("./rutas/ordenesRuta")

const PORT = process.env.PORT || 8000
const app = express()

app.use(express.json())

app.use(ENDPOINTS.MENU, menuRutas)
app.use(ENDPOINTS.MESAS, mesasRutas)
app.use(ENDPOINTS.MESEROS, meserosRutas)
app.use(ENDPOINTS.TIPOS_SERVICIO, tiposServicioRutas)

app.use(ENDPOINTS.ORDENES, ordenesRutas)

app.listen(PORT, () => {
  console.log(`Server listening on PORT: ${PORT}`)
})
```

*Figura 45. Servidor del microservicio Restaurante (Elaboración propia)*

### 4.4 Microservicio Analítica de datos

Este microservicio se encarga de obtener la información de las diferentes tablas del microservicio anterior para hacer consultas que permitan la generación de reportes, tiene similar estructura de carpetas salvo por los *utils* que para este caso no fueron necesarios, posee una carpeta adicional denominada *modelos,* y se diferencia principalmente del microservicio *Restaurante* dado que debe ejecutar una conexión a la base de datos documental descrita anteriormente, además de requerir una dependencia adicional, se desplegó en *Railway* un proyecto alojado en un repositorio de github, desarrollado en *NodeJS,* con la estructura de carpetas evidenciada en la *Figura 46*:

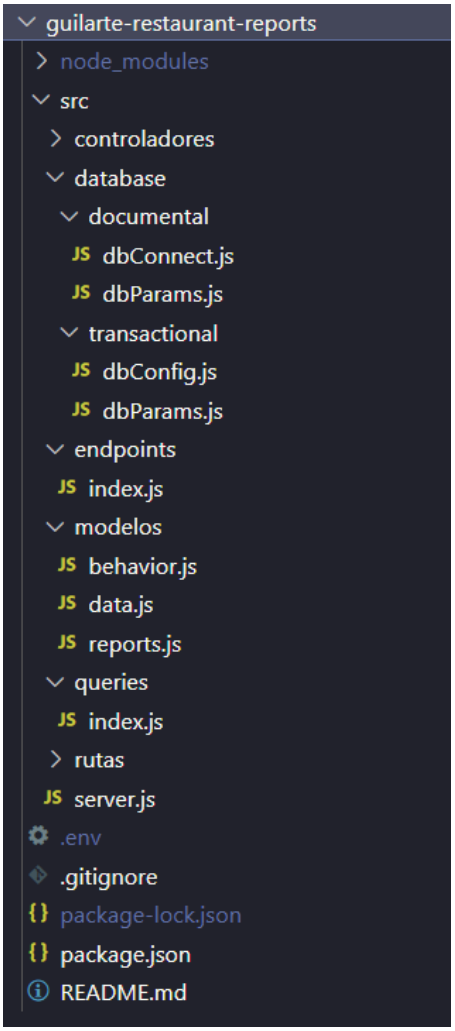



*Figura 46. Estructura de carpetas microservicio Reportes (Elaboración propia)*

Este proyecto cuenta con las mismas dependencias del anterior microservicio salvo por la siguiente:

- *mongoose:* Es una librería que funciona como *ODM (Object Document Model)* para *NodeJS* que permite la conexión y realización de consultas a bases de datos *MongoDB*.

A continuación se profundiza un poco sobre las carpetas de la *Figura 46:*

- ***database:*** Posee dos carpetas internas:
    - ***transactional:*** Consta de los siguientes archivos:
        - *dbConfig.js:* Archivo de conexión de la base de datos SQL.
        - *dbParams.js:* Archivo con los valores de conexión a la base de datos SQL y nombres de las respectivas tablas.

- ***documental:*** Consta de los siguientes archivos:
    - *dbConnect.js:* Archivo de conexión de la base de datos NoSQL.

```
require("dotenv")
const mongoose = require ("mongoose")

function documentalConnection() {
  const mongoURI = process.env.MONGO_DB
  mongoose.connect(mongoURI, {
    useNewUrlParser: true,
    useUnifiedTopology: true,
  })

  mongoose.connection.once("open", () =>
    console.log("Connected with Restaurante Guilarte Documental DB")
  )

  mongoose.connection.on("error", (err) =>
    console.log(`Something went wrong ${JSON.stringify(err)}`)
  );
}

module.exports = { documentalConnection }
```

*Figura 47. Archivo de configuración conexión base de datos documental MongoDB (Elaboración propia)*

    - *dbParams.js:* Archivo con los valores de conexión a la base de datos NoSQL y nombres de las respectivas tablas.

- ***controladores:*** Se encuentran las funciones que el servidor ejecutará cuando se llame a una de las rutas, tal cómo en el microservicio *Restaurante*, incluyendo las acciones para el guardado de cada uno de los documentos en sus respectivas colecciones.

```
const pool = require("../database/transactional/dbConfig")
const { REPORTS } = require("../database/documental/dbParams")

const Behavior = require("../modelos/behavior")
const Data = require("../modelos/data")
const Report = require("../modelos/reports")

const { QUERIES } = require("../queries")

module.exports = {
    async TOP_MENU_TYPES (req, res) {
        try {
            const { query: { startDate, endDate } } = req
            const topTiposMenu = await pool.query(QUERIES.GET_TOP_MENU_TYPES(startDate, endDate))
            const mostConsumed = topTiposMenu[0]
            const lessConsumed = topTiposMenu[topTiposMenu.length - 1]
            const saveMost = await Data.create({
                name: mostConsumed[REPORTS.MENU_TYPES.REPORTS.TYPES.XAXIS],
                total: mostConsumed[REPORTS.MENU_TYPES.REPORTS.TYPES.YAXIS],
            })
            const saveLess = await Data.create({
                name: lessConsumed[REPORTS.MENU_TYPES.REPORTS.TYPES.XAXIS],
                total: lessConsumed[REPORTS.MENU_TYPES.REPORTS.TYPES.YAXIS],
            })
            const saveReport = await Report.create({
                start_date: startDate,
                end_date: endDate,
                evaluated: topTiposMenu.length,
                most: [saveMost],
                less: [saveLess],
            })
            await topTiposMenu.forEach(async item => {
                const saveTop = await Data.create({
                    name: item[REPORTS.MENU_TYPES.REPORTS.TYPES.XAXIS],
                    total: item[REPORTS.MENU_TYPES.REPORTS.TYPES.YAXIS],
                })
                const currentReport = await Report.findById(saveReport._id)
                currentReport.top.push(saveTop)
                await currentReport.save()
            })
            const fullCurrentReport = await Report.findById(saveReport._id)
            const behavior = await Behavior.findOne({ name: REPORTS.MENU_TYPES.REPORTS.TYPES.NAME })
            behavior.reports.push(fullCurrentReport)
            await behavior.save()
            return res.status(200).json({
                results: topTiposMenu,
                xAxis: REPORTS.MENU_TYPES.REPORTS.TYPES.XAXIS,
                yAxis: REPORTS.MENU_TYPES.REPORTS.TYPES.YAXIS,
            })
        } catch (err) {
            return res.status(400).json({
                mensaje: "Hubo un error con la petición",
                error: JSON.stringify(err),
            })
        }
    },
}
```

*Figura 48. Ejemplo controlador de tipos de servicio (Elaboración propia)*

- ***rutas:*** Hace referencia a las petición *HTTP* que debe ejecutarse en determinado *endpoint* y la función del controlador que debe ejecutar.

```
const router = require("express").Router()
const tiposMenuController =
require("../controladores/tiposMenuControl")

const { ENDPOINTS } = require("../endpoints")

router.route(ENDPOINTS.BASE).get(tiposMenuController.TOP_MENU_TYPES)
,

module.exports = router
```

*Figura 49. Ejemplo rutas para el menú (Elaboración propia)*

- ***endpoints:*** Es el archivo donde se encuentran las variables para cada uno de los endpoints.
- ***modelos:*** Se encuentran las estructuras de cada una de las colecciones descritas en el diseño de la base de datos documental.

```
const { model, Schema } = require('mongoose')

const dataSchema = new Schema({
    name: {
        type: String,
        default: "",
        required: true,
    },
    total: {
        type: Number,
        default: 0,
        required: true,
    },
})

const Data = model('Data', dataSchema)

module.exports = Data
```

*Figura 50. Ejemplo modelo colección datas (Elaboración propia)*

- ***queries:*** Es el archivo donde se encuentran todas las queries a ejecutarse, dependiendo de cual sea la llamada dentro del archivo de controladores.

```
const { TABLE_NAMES } = require("../database/transactional/dbParams")
const { MENUS, MENU_TYPES, CUSTOMERS, ORDERS, ORDERS_MENUS } = TABLE_NAMES

module.exports = {
    QUERIES: {
        GET_TOP_MENU_TYPES: (startDate, endDate) =>
            `SELECT tm.nombre_tipo_menu, COUNT(tm.id_tipo_menu) AS "Total Vendido" FROM ${ORDERS} os
                JOIN ${ORDERS_MENUS} osm ON os.id_orden = osm.id_orden
                JOIN ${MENUS} m ON osm.id_menu = m.id_menu
                JOIN ${MENU_TYPES} tm ON tm.id_tipo_menu =m.id_tipo_menu
                WHERE os.fecha_orden BETWEEN "${startDate} 00:00:01" AND "${endDate} 23:59:59"
                GROUP BY tm.id_tipo_menu
                ORDER BY COUNT(tm.id_tipo_menu) DESC`,
    },
}
```

*Figura 51. Archivo de consulta Top tipos de menú a la base de datos (Elaboración propia)*

El archivo más importante en este microservicio es el *server,js* que se encarga de crear el servidor y conectar las rutas.

```
require("dotenv").config()
const express = require("express")

const { documentalConnection } = require("./database/documental/dbConnect")

const { ENDPOINTS } = require("./endpoints")
const menuRutas = require("./rutas/menuRuta")
const clienteRutas = require("./rutas/clienteRuta")
const tiposMenuRutas = require("./rutas/tiposMenuRuta")

const PORT = process.env.PORT || 8000
const app = express()
documentalConnection()

app.use(express.json())

app.use(ENDPOINTS.TOP_MENUS, menuRutas)
app.use(ENDPOINTS.CUSTOMERS, clienteRutas)
app.use(ENDPOINTS.MENU_TYPES, tiposMenuRutas)

app.listen(PORT, () => {
  console.log(`Server listening on PORT: ${PORT}`)
})
```

*Figura 52. Servidor del microservicio Reportes (Elaboración propia)*

# PARTE III. CIERRE DE LA INVESTIGACIÓN

# CAPÍTULO 5. CONCLUSIONES

## 5.1 Verificación, contraste y evaluación de los objetivos

### *5.1.1. Verificación de Objetivos*

De acuerdo a los objetivos propuestos al inicio de la investigación y en relación con el desarrollo y diseño del prototipo propuesto, se realizó una verificación de cada uno de los objetivos que dan resultado a la posterior evaluación. El prototipo de la herramienta tecnológica desarrollado proporciona a través de técnicas de análisis de datos gráficos que facilitan la toma de decisiones de acuerdo a la inteligencia de negocios en el restaurante Guilarte de la ciudad de Bogotá. Esta herramienta está conformada por un *backend* con microservicios conectados a un *API (interfaz de programación de aplicaciones)* que vinculan la capa de almacenamiento y la interfaz de usuario en el *frontend* para que el usuario final

pueda visualizar la información allí mostrada y de esta manera tomar decisiones basadas en la proyección que se tiene para la inteligencia de negocio del restaurante.

### *5.1.2. Evaluación de objetivos.*

Para poder evaluar los objetivos específicos, primero, se sugirió en la *sección 3.2.1* un modelo de base de datos basado en la lógica de negocio del restaurante y definido bajo los requerimientos levantados al inicio de la investigación. Esta base de datos responde a la necesidad de almacenar información relevante para posteriormente aplicar herramientas de analítica de datos. En segunda instancia, se diseñó un algoritmo a través de consultas a la base de datos, este algoritmo es capaz de visualizar indicadores que según los datos permite analizar variables de consumo en un período de tiempo determinado por el usuario. Finalmente, estos datos son representados en la interfaz de usuario mediante gráficas que permiten la toma de decisiones para futuros eventos previstos en el restaurante.

## 5.2 Síntesis del modelo propuesto

El modelo propuesto que permite llegar a los resultados finales de la investigación parte de la sugerencia de un modelo de lógica de negocio el cual propone el control de los menús de las órdenes de servicio de los clientes y de esta manera propicia el paso a la creación del modelo de la base de datos que satisface las necesidades de los objetivos específicos. Posteriormente, con base en el modelo de la base de datos, se propuso una arquitectura de software donde los microservicios involucran el *backend* con el *frontend* para que de esta manera se puedan visualizar los datos con información oculta importante para la toma de decisiones y en consecuencia sean almacenados en una base de datos documental diseñada en la *sección 3.2.2.*

La interfaz propuesta cuenta con dos módulos, órdenes de servicio y Reportería, tal como se muestra en la *Figura 53*

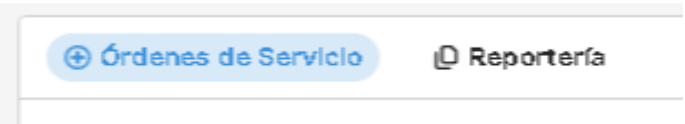



*Figura 53. Módulos Interfaz de usuario (Elaboración propia)*

En el módulo de Órdenes de servicio, se encuentra un formulario el cual contiene tres segmentos donde el usuario ingresará información para completar la órden de servicio. Todos estos campos son obligatorios. En la *Figura 54* se muestra dicho formulario.

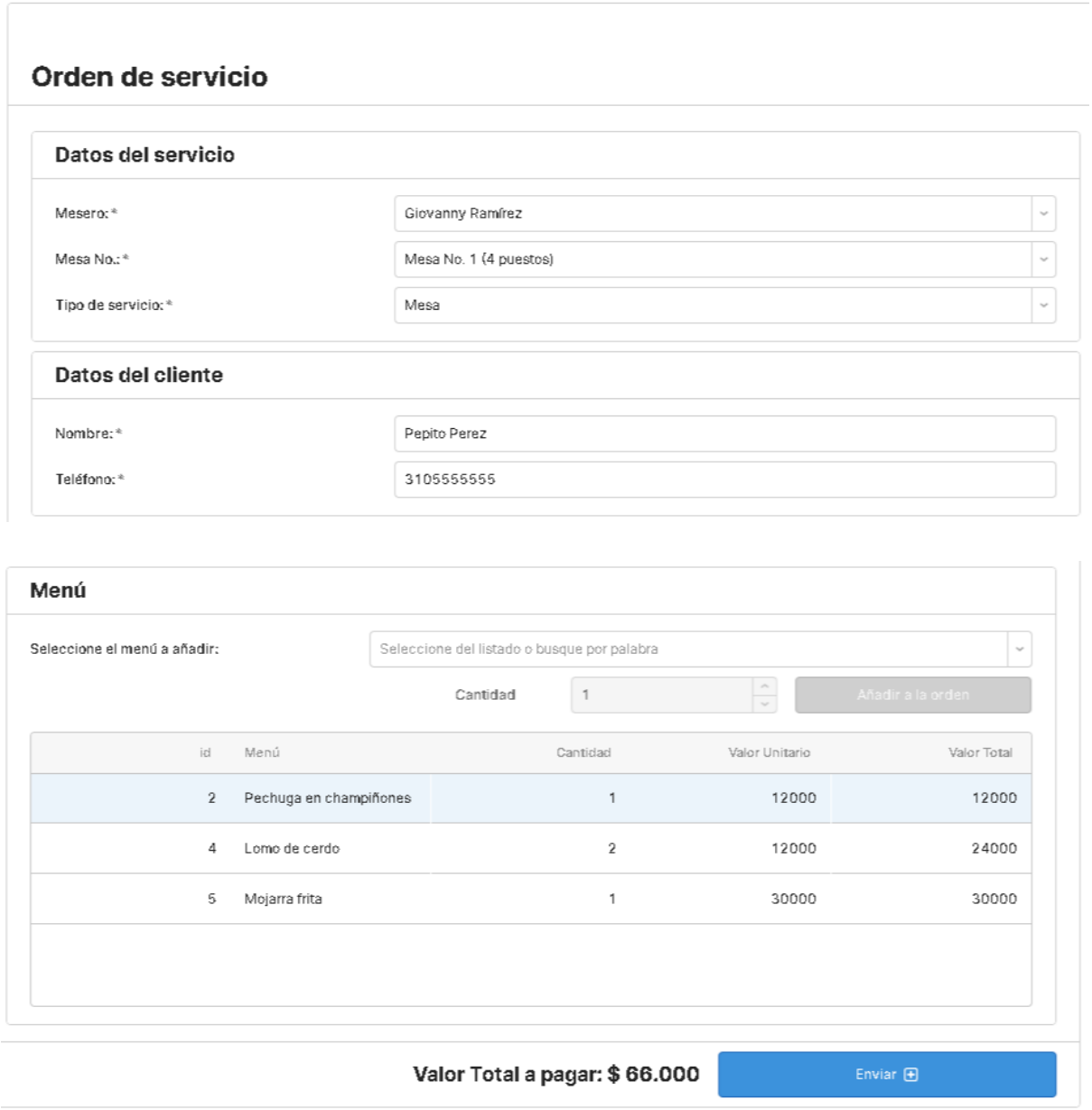



*Figura 54. Módulo Órdenes de servicio (Elaboración propia)*

Una vez se hace el envío del formulario, se realiza una petición al microservicio *Restaurante* con los datos suministrados, creando una orden de servicio y consignando los campos en la base de datos.

El segundo módulo llamado Reportería, tiene la funcionalidad de generar reportes según las consultas establecidas, para de esta manera visualizarlos en el frame, tal como se muestra en la *Figura 55*.




*Figura 55. Módulo Reportería (Elaboración propia)*

### 5.3 Resultado de Reportes

El módulo de Reportería genera seis (6) reportes diferentes, donde el parámetro principal es el rango de fechas del cual se desea obtener información.

- ***Top menús:*** genera un reporte indicando la cantidad de menús vendidos en un rango de fecha determinado, tal como se muestra en la *Figura 56,* donde el menú más vendido en ese rango de fecha es la *Mojarra Frita* con un total de 16 unidades vendidas mientras que Churrasco asado es el menú menos vendido con un total de 2 unidades vendidas.

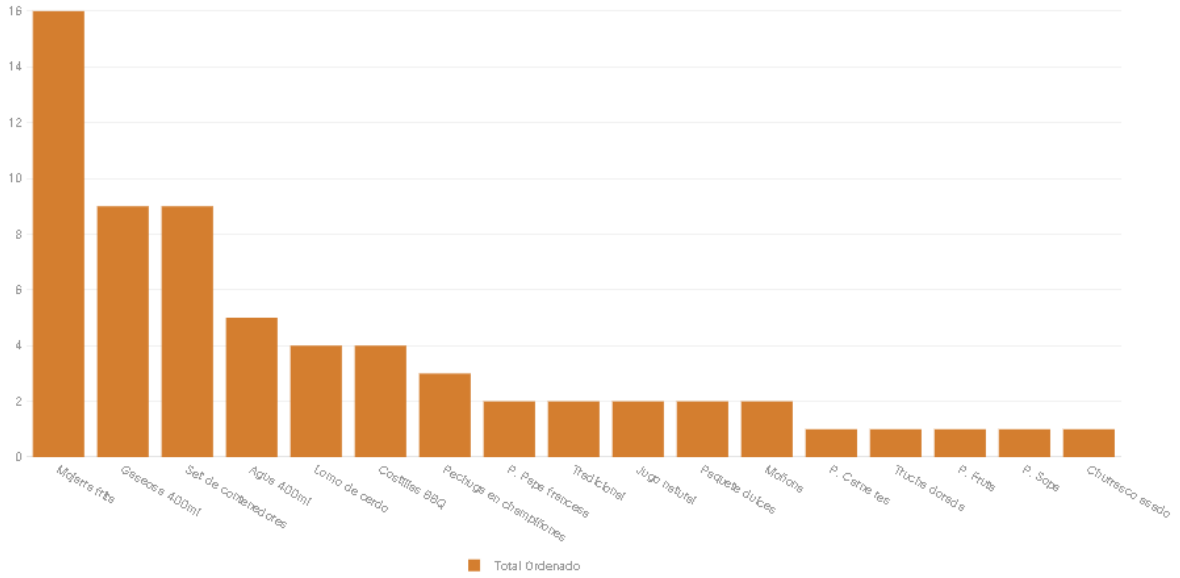



*Figura 56. Reporte "Top menús" (Elaboración propia)*

- ***Top menús valorados:*** Permite la visualización del top de menús que más generan ingresos en el sistema, una vez más para el caso de ejemplo, la *Mojarra Frita* es la que más genera ingresos superando los *$400.000* en venta, mientras que la *Porción de Sopa* es el producto que menos genera ingresos por debajo de los $100.000 pesos, tal como se muestra en la siguiente figura.

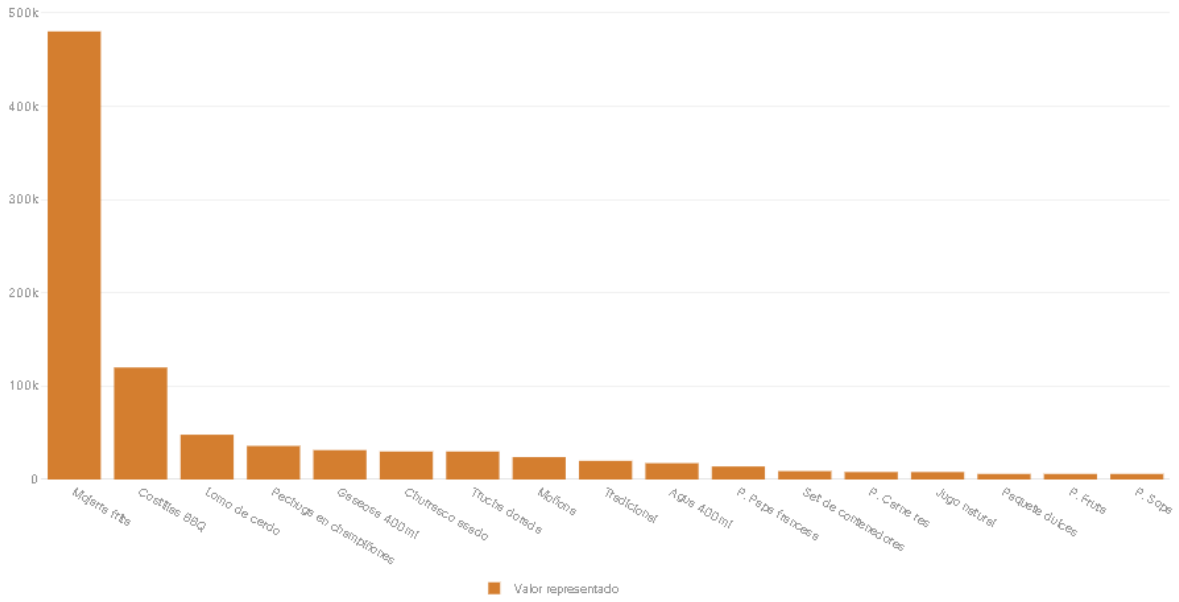



*Figura 57. Reporte "Top más valorados" (Elaboración propia)*

- ***Top menús por cliente:*** Devuelve el récord de los menús más solicitados por un cliente específico de acuerdo a su número telefónico (que es el identificador principal del cliente) durante un periodo de tiempo específico, tal como se muestra en la siguiente figura.

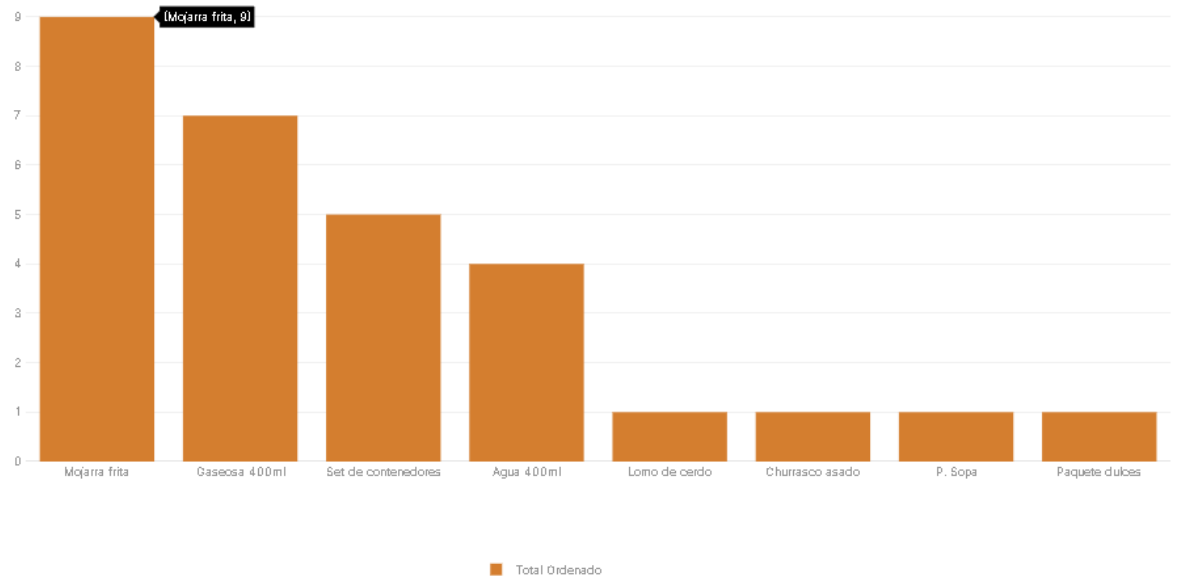



*Figura 58. Reporte "Top menús por cliente" (Elaboración propia)*

Teniendo en cuenta el modelo de lógica de negocio para el restaurante Guilarte, se decidió añadir el reporte:

- ***Top tipos de menú más vendidos:*** Donde se puede apreciar el tipo de menú que más se ha vendido en un determinado periodo de tiempo, así como se muestra en la *Figura 59*, la cual muestra que la venta de *Almuerzo Especial* es más vendido que otros tipos de menús ofrecidos por el restaurante.

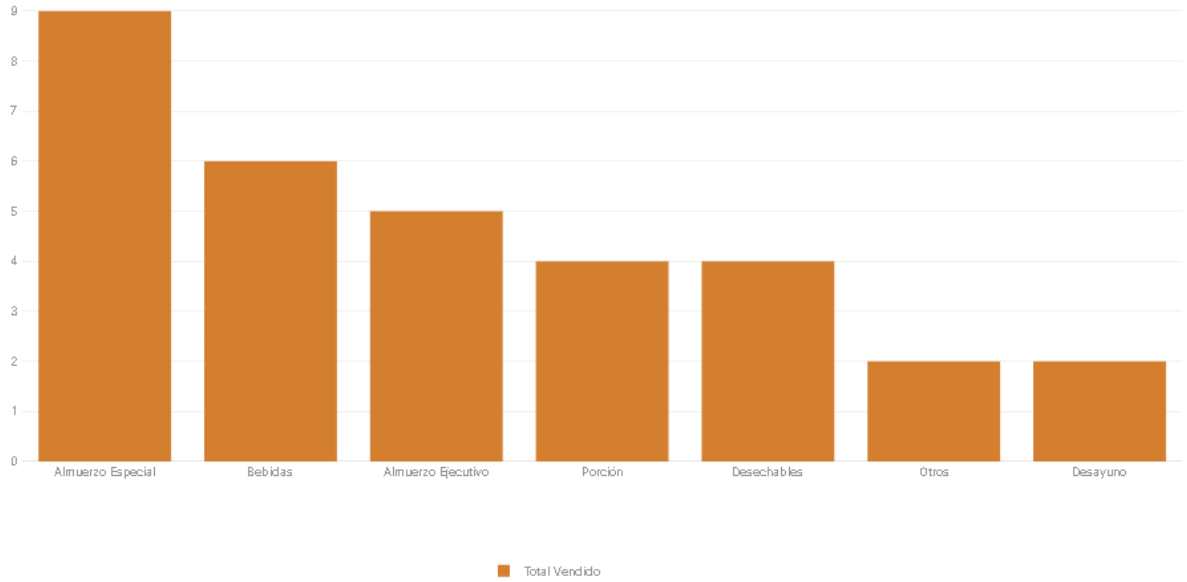



*Figura 59. Reporte "Top tipos de menú más vendidos" (Elaboración propia)*

Pensando en un futuro plan de fidelización a clientes recurrentes:

- ***Cliente que más ha gastado:*** Retorna los *n* puestos de clientes que más han consumido en el restaurante, tal como lo muestra la *Figura 60,* donde el cliente “Giovanny Ramirez” ha gastado casi $1.800.000 pesos en compras en el restaurante.

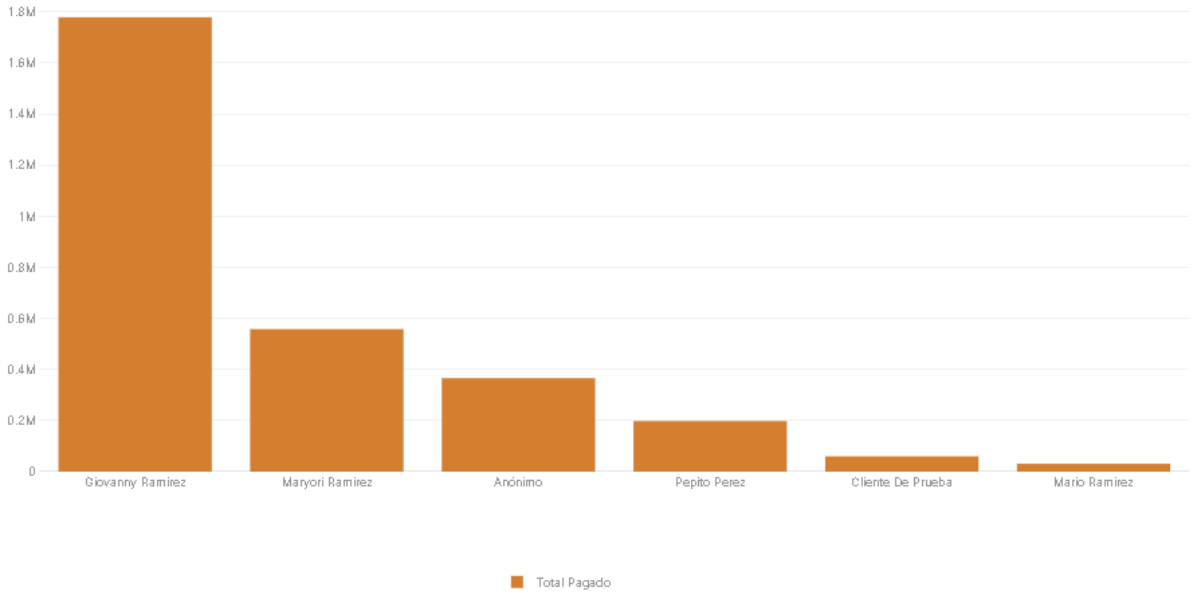



*Figura 60. Reporte “Cliente que más ha gastado” (Elaboración propia)*

El último de los reportes que se pueden generar en el este modulo de Reportería es:

- ***Cliente con más órdenes de servicio:*** Hace referencia al cliente con la mayor cantidad de órdenes de servicio realizadas en un periodo de tiempo determinado, por ejemplo, en la *Figura 61* se observa que un cliente “Anónimo” tiene 7 órdenes de servicio, una aclaración importante en este punto es que el cliente anónimo representa a todos aquellos que por privacidad prefieren no dar su número de teléfono, por lo que en el formulario de las órdenes de servicio, no importa el valor del nombre ingresado en el campo si el valor del campo teléfono es cero (0).

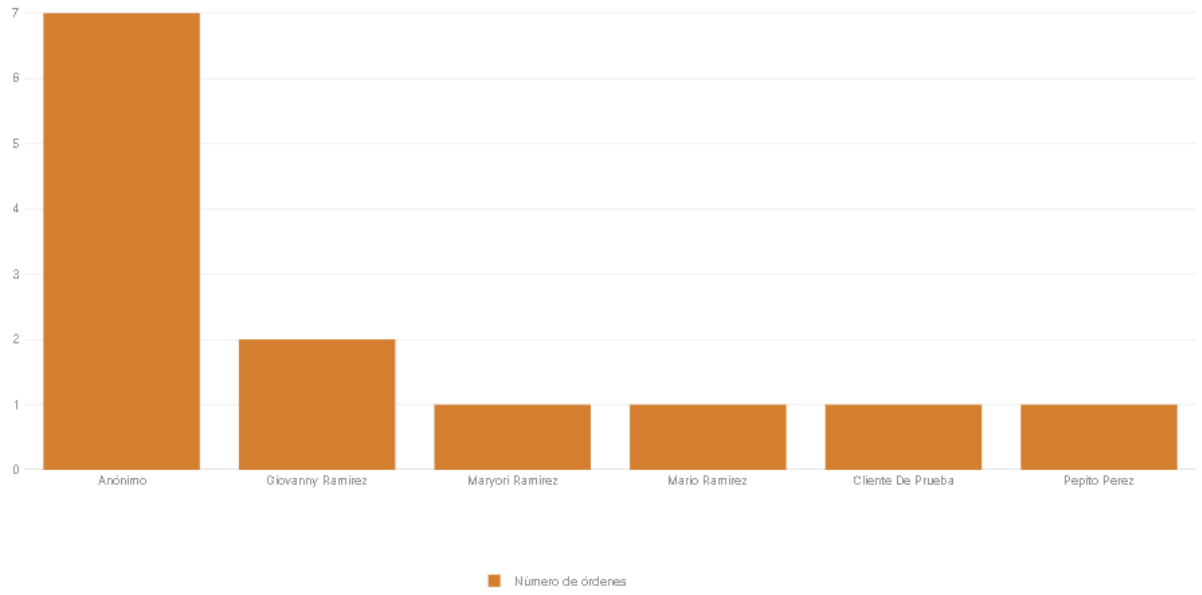



*Figura 61. Reporte "Cliente con más órdenes de servicio" (Elaboración propia)*

### 5.4 Aportes originales

Diseño y Arquitectura de un aplicativo web enfocado en análitica de datos aplicados a Restaurantes, basado en la inteligencia de negocios.

Implementación de una capa de servicios REST basada análitica de datos para la toma de decisiones en la inteligencia de negocio del restaurante Guilarte.

## CAPÍTULO 6. PROSPECTIVA DEL TRABAJO DE GRADO

### 6.1 Alcances y limitaciones

Esta implementación estuvo enfocada en determinar cómo la analítica de datos puede ser utilizada para mejorar la experiencia de los clientes, también cómo desde la administración del restaurante se puede monetizar el conocimiento de los datos y generar mayor rentabilidad. El resultado más relevante para esta investigación fue mostrar la implementación del modelo de inteligencia de negocios donde la información que es susceptible a ser procesada pueda brindar sugerencias respecto a la toma de decisiones que hacen que el negocio crezca previendo los posibles eventos negativos que impacten el nicho del mismo o prediciendo la cantidad de menús programados para una fecha en específico teniendo en cuenta la tendencia de los datos.

## 6.2 Líneas de investigación futuras

A. Integración módulos de inteligencia artificial para predicción de comportamientos y tendencias.
B. Integración de tablas con cada uno de los ingredientes de los menús.
C. Integración módulos para creación y edición de clientes y menús.

**BIBLIOGRAFÍA**

*Business Intelligence*. (2022). CMI Gestión. Retrieved Mayo 31, 2022, from https://cmigestion.es/business-intelligence/

Conexión ESAN. (2019, Marzo 7). *¿Qué es Data Mart y por qué es importante implementarlo en tu empresa?* | *Conexión ESAN*. ESAN. Retrieved Mayo 31, 2022, from https://www.esan.edu.pe/conexion-esan/que-es-data-mart-y-por-que-es-importante-implementarlo-en-tu-empresa

Covarrubias, M., & Garza, K. (n.d.). *La importancia del inventario en tu restaurante y cómo realizarlo*. Blog | CookingDepot.com. Retrieved Mayo 18, 2022, from https://blog.cookingdepot.com/la-importancia-del-inventario-en-tu-restaurante-y-como-realizarlo

DBA EXPERTS TECH. (2021, Mayo 19). *Lo más importante en un proyecto de Analítica de Datos – DbaExperts*. DBA Experts. Retrieved Mayo 31, 2022, from https://dbaexperts.tech/wp/database/proyecto-de-analitica-de-datos/

Diccionario Actual. (2022). *¿Qué es big data? Definición, concepto y significado.* Diccionario Actual. Retrieved Mayo 31, 2022, from https://diccionarioactual.com/big-data/

Emérita Legal. (2018, Agosto 30). *El Machine Learning cambiará el mundo*. Emérita Legal. Retrieved Mayo 31, 2022, from https://www.emerita.legal/blog/innovacion-legal/el-machine-learning-cambiara-el-mundo/

Fundeu RAE. (2013, Febrero 7). *«macrodatos» e «inteligencia de datos», alternativas a «big data»*. Fundéu. Retrieved Mayo 31, 2022, from https://www.fundeu.es/recomendacion/macrodatosalternativa-abig-data-1582/

González, L. (2021, Abril 13). *Reglas de Asociación - Aprende IA*. Aprende IA. Retrieved Mayo 31, 2022, from https://aprendeia.com/reglas-de-asociacion/

Gutiérrez Reina, D., & Rodríguez del Nozal, Á. (2020). *Algoritmos Genéticos con Python: un enfoque práctico para resolver problemas de ingeniería*. Marcombo.

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*. Wiley. chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://aatinegar.com/wp-content/uploads/2016/05/Kimball_The-Data-Warehouse-Toolkit-3rd-Edition.pdf

Lee, M., Kwon, W., & Back, K.-J. (2021). Artificial intelligence for hospitality big data analytics: developing a prediction model of restaurant review helpfulness for customer decision-making. *International Journal of Contemporary Hospitality Management*, *33*(6). https://doi-org.bdigital.udistrital.edu.co/10.1108/IJCHM-06-2020-0587

Martinez Garcia, J. H. (2010). *La inteligencia de negocios como herramienta para la toma de decisiones estratégicas en las empresas. Análisis de su aplicabilidad en el contexto corporativo colombiano.* Universidad Nacional De Colombia. Retrieved Mayo 31, 2022, from http://repositorioslatinoamericanos.uchile.cl/handle/2250/580990

Menzies, T., & Hu, Y. (2003, Noviembre). Data Mining for Very Busy People. *IEEE Computer Society*, *36*(11), 22-29. 10.1109/MC.2003.1244531

Microsoft. (2022). *Procesamiento analítico en línea (OLAP) - Azure Architecture Center*. Microsoft Docs. Retrieved Mayo 31, 2022, from https://docs.microsoft.com/es-es/azure/architecture/data-guide/relational-data/online-analytical-processing

Peña, D., & De Rivera, D. P. S. (2010). *Análisis de series temporales*. Alianza Editorial.

*Qué es un almacén de datos*. (2022). Oracle. Retrieved Mayo 31, 2022, from https://www.oracle.com/co/database/what-is-a-data-warehouse/

*¿Qué son las redes neuronales y cómo funcionan?* (2022, Enero 19). Universidad Central. Retrieved Mayo 31, 2022, from https://www.ucentral.edu.co/noticentral/redes-neuronales

Rodrigo, J. A. (2018, Junio). *Reglas de asociación y algoritmo Apriori con R*. Cienciadedatos.net. Retrieved Mayo 18, 2022, from https://www.cienciadedatos.net/documentos/43_reglas_de_asociacion

Russell, S. J., Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. Pearson.

Slaughter, R. (1995). *Futures for the Third Millenium: Enabling the Forward View*. Prospect Media.

Swink, M., Hu, K., & Zhao, X. (2022). Analytics applications, limitations, and opportunities in restaurant supply chains. *Production & Operations Management*, *1*. https://doi-org.bdigital.udistrital.edu.co/10.1111/poms.13704

*10 técnicas de análisis de datos para estadísticas de big data*. (2017, Diciembre 5). Gestión de Datos. Retrieved Mayo 31, 2022, from https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/10-tecnicas-de-analisis-de-datos-para-estadisticas-de-big-dat